Automating Symptom Information Extraction from the Electronic Health Record: A Context Aware Deep Learning Approach

Jonathan S. Fan [a], Min Zhang [b], Ling Tong [c], Alaa AlBashayreh [d], Weiguo Fan [c], Stephanie Gilbertson-White [d]

[a] Iowa City West Senior High School, Iowa City, IA, United States

[b] Informatics, University of Iowa, IA, United States

[c] Department of Business Analytics, University of Iowa, IA, United States

[d] College of Nursing, University of Iowa, IA, United States

Abstract

Extracting symptom information from the electronic health record (EHR) is a task that healthcare providers and researchers regularly deal with. Traditionally, symptom extraction is a manual search process that costs time and human efforts. With the increasing use of EHR data in routine hospital encounters and clinical visits, evidence suggests that healthcare providers are at a higher risk for EHR burden and burnout. This is true, especially during the COVID-19 pandemic. In fact, time spent on analyzing EHR data increased by 157% compared to the pre-pandemic average. How to quickly identify patient symptom information from EHRs is thus of paramount importance for increasing healthcare providers productivity and reducing the burnout rate. In this research, we propose a deep learning-based framework that combines customized word embedding techniques and deep neural networks for symptom extraction. Using the Medical Information Mart for Intensive Care (MIMIC-II) Database, we experiment various models with different combinations of word embeddings plus a bidirectional long short-term memory (biLSTM) neural network to automate symptom extraction. The best-performing model achieves an F1 score of 0.956 using a pre-trained GLoVe embedding concatenated with a self-trained FastText embedding plus a biLSTM neural network classifier. This model efficiently extracts symptom information and outperforms the benchmarks by more than 10%. Adopting such a fast and accurate system can potentially reduce EHR burden and improve healthcare and research.

Keywords
Healthcare Informatics; Symptom Extraction; Deep Learning; Text Mining; Word Embedding

1. Introduction

Currently, healthcare provider burnout rates are increasing and becoming more of an impending problem due to the COVID-19 pandemic and its rapid increasing rates (Holmgren et al., 2021); new innovative solutions are required to solve this issue. One of the leading reasons more healthcare providers are experiencing burnout is due to the increasing use of electronic health records (EHR) that providers must go through during hospital encounters and clinic visits. Research indicates that there is a strong correlation between healthcare provider burnout and the time required to thoroughly analyze an EHR (Patel et al., 2018; Shih et al., 2013). Healthcare providers who spent more than a median of 6 hours per week on EHR work were nearly 3 times more likely to report burnout than those who spent less time (Robertson et al., 2017). In addition, the usability scale for EHRs averaged a grade of "F" over published surveys of nearly 900 physicians (Sutton et al., 2019), which suggests that EHRs are extremely difficult to read and digest quickly. Healthcare providers must analyze many lines of unstructured text to extract key information, while also disregarding inaccurate, dispensable, out of date information within each EHR.

To reduce EHR burden and burnout rates among healthcare providers and efficiently identify key information (e.g., patient symptoms, medications, medical diagnoses and recommendations), an automatic information extraction (IE) tool is needed. By utilizing an IE tool, the burden of data retrieval of key information and summarization within an EHR shifts from the healthcare provider to the software, thereby reducing the burnout rates of healthcare providers. In addition, IE tools would be able to extract crucial information more accurately from EHRs than a human abstractor if properly utilized and trained (Liu & Kauffman, 2020). In this work, we specifically focus on the task of symptom extraction from EHRs. It is known that EHRs contain vital information that healthcare providers must accurately identify to provide effective treatment, such as medications, diagnoses, treatment plans, and symptoms (Kalra, 2006) - with symptoms being the most crucial

information for medical diagnosis (Holmgren et al., 2021). Symptom extraction is the process of recognizing and extracting words that describe a patient's symptoms from his or her EHR. We seek to design a high-performing symptom extraction IE system that would allow healthcare providers to efficiently extract symptom information, thereby reducing EHR-burden and burnout rates. For example, an IE system could extract *"lower back pain"* from *"the patient is experiencing lower back pain"* and categorize the phrase as a symptom.

Creating an IE tool to automate this extraction task is very challenging. First, symptoms can be mentioned in many different sections of the EHR, but not all of them are indeed experienced and described by patients. For instance, symptoms can be documented as medication indications or adverse reactions for prescribed medicine. Also, they can be mentioned in a patient's family history. Moreover, symptoms can be mentioned in the teaching tips documented in the EHR. We follow Steinkamp et al. (2020) to define a symptom as "a phenomenon subjectively experienced by a patient or observed by either the patient, or another person (e.g., family member)." If a patient describes experiencing "chest pain", then "chest pain" would be considered as symptom words. However, if "chest pain" is mentioned as medication indications, adverse reactions or exam findings, it would not be considered as symptom words in this research. Second, healthcare providers may use different terminologies to describe the same symptom, for example, "chest pain" might be abbreviated as "CP". Finally, the proportion of symptom words is very small in the EHR, which creates a highly imbalanced classification problem, that is, symptom mentions would be a minority class compared to no symptom mentions. Many IE systems are currently available; however, these systems rely on either a rule-based or a dictionary-based approach to extract information (Oyelade et al., 2018). A rule-based or a dictionary-based approach would identify all occurrences of predefined words, regardless of whether they are truly experienced by patients. Also, these systems would only work within the given rules and are incapable of generalizing beyond. For example, new symptom words that show up after the rules are defined would not be able to recognize by such systems.

In this research, we develop an IE system that can effectively extract symptom words and is free of the rule- and dictionary-based restrictions, using natural language processing (NLP) algorithms and deep learning (DL) models in Artificial Intelligence (AI). Specifically, we use word

embeddings coupled with bidirectional long short-term memory (biLSTM) neural networks. Unlike rule- and dictionary-based approaches, our deep learning model extracts symptom words based on semantic and contextual information embedded in the EHRs. Therefore, our model is capable of differentiating between true symptoms experienced and reported by patients versus other symptoms. Moreover, the model has generalizability compared with the rule- and dictionary-based approaches, as it can recognize different terminology in describing the same symptom and new symptom words that are not included in the training process.

The rest of our paper is structured as follows: we first introduce deep learning techniques that will be applied in this study and examine current research methods utilized to extract symptom data in Section 2. Section 3 details the framework proposed in this paper for symptom extraction, including the methodology tested to achieve the best symptom extraction performance such as data preprocessing and deep learning experimentation. Section 4 analyzes the results of each benchmark and provides the most effective IE system. We then discuss future implications for research, potential applications with our model, and limitations that need addressing in Section 5. Section 6 concludes our paper with an overall summary of the highlights of our research.

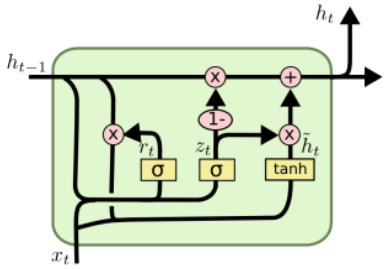
2. Related works and conceptual backgrounds

Our research is related to deep learning, contextual modeling through word embeddings, and information extraction within NLP processes. We will review each of these related works below.

2.1 Deep learning

Deep learning allows multiple processed layers within a computation model to learn various representations of data with increased abstraction (LeCun et al., 2015). Primary NLP algorithms incorporate deep learning approaches such as recurrent neural networks (RNNs) to negate the limitations provided from a lexicon approach. RNNs are well-known for their capability of modeling sequential data. A more specific type of RNNs are LSTMs networks (Hochreiter & Schmidhuber, 1997). LSTMs are cell states that recurrently undergoe nonlinear transformations on a piece of data. In this example below, the first step in an LSTM model is to determine what information will be discarded within a cell state: $z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$. The decision is made

through a sigmoid layer (*"forget gate layer"*) containing x_t and h_{t-1}, which outputs a number between 0 (*"completely get rid of"*) and 1 (*"completely keep this"*). The next step is to determine new information that will be stored in each cell state. First, a sigmoid layer (*"input gate layer"*) decides which values will be updated: $r_t = \sigma(W_i \cdot [h_{t-1}, x_t])$. Next, a tanh layer creates a vector of new related candidate values that could be added to each cell state: $\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$. Finally, the last nonlinear transformation determines where each output will go: $h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$ (Olah, 2015). Afterwards, each output will be passed to the next piece of data, demonstrating the power of the LSTM architecture to produce key labels and dynamic length inputs. Furthermore, biLSTM creates a sequence processing model that consists of two LSTMs: one taking the input in a forward direction, and the other in a backwards direction. These two LSTM outputs are then averaged in an element-wised fashion to produce an output vector that is then passed onto later LSTM units. By applying biLSTM layers into our model, we can capture more complex and comprehensive relationships between words and create an IE system that is able to generalize beyond a given set of rules.



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$
$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Figure 1 LSTM Unit Calculations[1]

2.2 Contextual modeling through word embeddings and transformers

Word embeddings are the method of converting a set of words into an $N(number\ of\ words) \times D(dimensional\ output)$ matrix. One of the most popular traditional word embeddings is the Word2Vec algorithm (Mikolov et al., 2013). Mikolov et al. experimented with an unsupervised learning method on untagged data to predict a word given its preceding context and a specific word input. This method outputs converted words into associated vectors to achieve predictions. Through this research, multiple large-scale word embeddings were created through training

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs

Word2Vec algorithms on large datasets such as Twitter and Wikipedia. Like Word2Vec, GloVe embeddings is a pre-trained word embeddings model that consists of a set of word vectors trained on over 5 billion articles from Wikipedia and other online data sources (Pennington et al., 2014).

Current research has introduced a new method of computing word embeddings through transformers. For instance, Vaswani et al., (2017) popularized transformers through the introduction of attention mechanisms and sequence-to-sequence algorithms. A transformer utilizes multi-headed attention mechanisms-a method of gathering context of a word, paired with a feed forward neural network, rather than a recurrent neural network, to create more efficient sequence-to-sequence tasks. Additionally, transformers consist of encoders and decoders; data is first "encoded" through encoders and then further "decoded" using the transformer's decoders. Within these decoders are multi-headed attention masks, meaning that all similarities between words and all information after a word are removed to ensure that preceding words are only considered for predictions. A specific type of transformers is the bi-directional encoder representations from transformers (BERT), which introduces novel mechanisms to bidirectionally train transformers to create more complex semantic relationships to evaluate connections between words at a sentence level (Devlin et al., 2018). Recently, more research has attempted to combine multiple different contextualized embeddings (e.g., concatenating BERT embeddings with GloVe embeddings). The combination of multiple different contextualized embeddings has shown to further improve the performance of downstream text mining tasks such as sentence classification sentiment prediction, and name entity recognition. (Zhang et al., 2021; Chang et al., 2021; Liu et al., 2020; Fan et al., 2020; Jianqiang Li et al., 2016; Zhai et al., 2019). Through experimenting with pre-trained embeddings like GloVe and current transformers such as BERT and different combinations, we can effectively develop a model that identifies greater context on a per-word basis during symptom extraction tasks.

2.3 Name entity recognition in information extraction

Our work is related to information extraction research, especially the work on named entity recognition (NER); the process of identifying known entities within a given piece of text such as time, people, organizations, symptoms, medications, and diseases (Mansouri et al., 2008). Deep

learning, a technique for NER, has been used extensively for concept extraction, information extraction, and information retrieval tasks. For example, Zhang et al. utilized a deep learning neural network to conduct name entity recognition and extract different accounts of the same news, to attribute certain information to a person (Zhang et al., 2019). Using deep learning, Zhang et al. (2019) outperformed previous machine learning models by approximately 11.96%, indicating the potential of deep learning in IE tasks. Feng et al. utilized natural language data augmentation combined with a biLSTM-CRF deep learning model in order to accurately conduct information extraction on a given report analysis to identify construction accidents (Feng & Chen, 2021). Within this study, the model trained on a limited data source yet still achieved reliable results, showing the power of deep learning to generalize and learn on a given set of text. Another paper (Hoang & Mothe, 2018) employs NLP techniques to detect geography-related terms within tweets. The NLP methods enabled the combination of information among various tweet metadata and extracted certain words that helped identify location within tweets. By proposing the problem of symptom extraction that is similar to information extraction tasks which use NER, namely extracting symptom keywords from a discharge summary, we can promptly develop a model using deep learning techniques to create functional IE systems that healthcare providers can use.

2.4 Symptom extraction from EHRs

Symptom extraction research has seen widespread growth within the medical report and electronic health record spectrum. Several methods have been proposed in the literature for symptom extraction from healthcare provider notes.

2.4.1 Lexicon-based approach

Standard lexicon-based approaches rely on the schematics of a given language. In symptom extraction research, lexicon-based extraction methods primarily use large data sets of medical terms. Some researchers have attempted to construct their own dictionaries, opting to create a vocabulary index based on phrase frequencies (Oyelade et al., 2018). Specifically, Oyelade et. al proposes the creation of an inference process, breast cancer lexicon, and rule set to extract symptom data relating to cancer. Oyelade et. al combined this dictionary with WordNet's natural language lexicon database to improve on information extraction within medical expert systems on

breast cancer diagnosing. This research demonstrates the validity of creating inferred lexicon indices from a given dataset; however, key limitations exist. Lexicon-based methods require constant updates and therefore may fail when new EHR formats are created, thereby extracting inaccurate symptom data for a healthcare provider to use. In addition, many concepts of language such as negation, severity, and prepositional phrases cannot be properly captured by a lexicon-based approach, meaning these models cannot generalize beyond the given set of rules applied to that model.

2.4.2 Deep learning approach

Due to the limitations of lexicon-based approaches, researchers have been exploring other alternative approaches using more advanced machine learning methods like deep learning. However, no work has attempted to analyze EHRs using a neural network combined with clinically trained embedding transformers to better extract symptom data. Therefore, the motivation behind our research is to improve on the current model approaches by incorporating novel deep learning features using biLSTM neural networks, pre-trained word embeddings, and self-trained/fine-tuned embedding models (Penning et al., 2014; Vaswani et al., 2017; Devlin et al., 2018).

3. Materials and Methods

The aim of this study is to automate symptom from the EHR discharge summary notes using deep learning to create an effective IE tool that healthcare providers could use, which has the potential to reduce EHR-burden and burnout rates.

3.1 Overview of framework

The objective of this framework is to take a discharge summary and extract words that are considered symptoms. The methodology is shown below in Figure 2 and described as follows. We take the data provided from the Medical Information Mart for Intensive Care (MIMIC-II) public dataset (Steinkamp et al., 2020) and utilize the data labels already provided within the data

to feed into our text processing method. Within this preprocessing method, we examine each discharge summary and mark all words with either 1 or 0 from the data labels provided. Before training, the data is converted into embeddings which are then fed into our biLSTM deep learning model. This model outputs probabilities for each word with a range of 0 to 1, where values closer to 1 means the word is more likely a symptom and values closer to 0 means the word is more likely not a symptom. Each box will be expanded upon in the following sections.

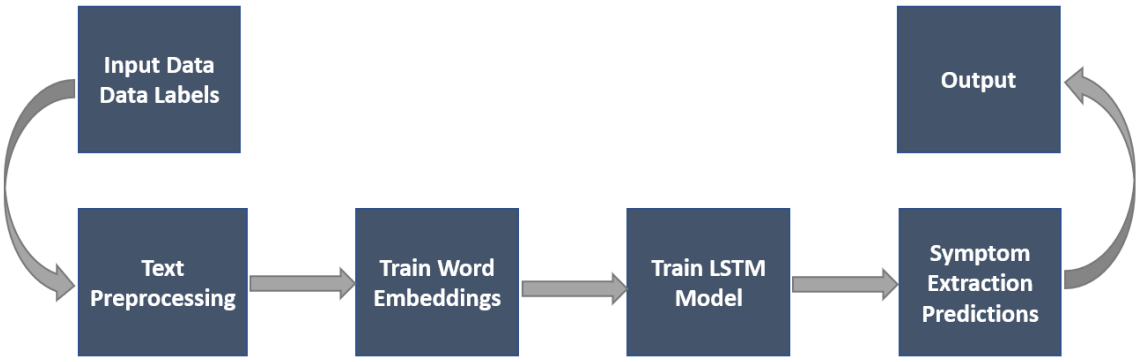


Figure 2. Proposed symptom extraction framework

3.2 Data

The dataset we analyzed is the MIMIC-II public dataset (n2c2 NLP Research Data Sets), which was originally developed during the i2b2 project (Steinkamp et al., 2020). In this dataset, there are 1,009 EHR discharge summaries and each summary was annotated by a medical student. The annotators followed an annotation guideline that utilized name entity recognition, name entity normalization, and coreference resolution to determine if a set of words was a symptom or not (Steinkamp et al., 2020). Though discharge summaries lack the representation of the full spectrum of clinical notes, they contain essential medical information such as lab findings, physical exams, assessments, and healthcare provider notes.

3.3 Task Definition

To begin identifying symptoms, we must first define what a symptom is. A symptom is an indication of a health-related concern such as pain, disturbed sleep, depressed mood, anxiety, cough, nausea, or dyspnea (Koleck et al., 2019). An example of symptoms within a discharge summary would be, "patient reported *lower back pain*". However, a symptom that is stated as part of medication prescription ("Take Tylenol for *headache*") or in a conditional statement ("If patient continues experiencing *chest pain*, prescribe medication") would not be considered a symptom, thereby creating a more difficult task of accurately extracting symptoms. In order to prevent string matching when identifying a symptom, deep learning must be applied in order to learn and analyze surrounding context and overall document structure to accurately identify a symptom. For this study, we evaluate the performance of our model to accurately identify these symptoms within a record file.

3.4 Data Preprocessing

To feed the data into our deep learning model, we must first process the raw discharge summaries and the human annotated indices of symptoms to create a suitable dataset. On average, each discharge summary consisted of 156 lines of clinical text information. To effectively use NER, we defined a sentence as one line from a given discharge summary. This would enable the deep learning model to effectively learn and use surrounding text to identify a symptom rather than feeding unrelated word-by-word tokens into the model. Furthermore, we divided each sentence into an array of tokens which had a classification of 0 or 1. If a given token was tagged 0, it was considered no symptom; a token tagged with 1 was considered a symptom. To determine if a token was a symptom or not, we utilized the human annotated indices of symptoms provided from the public dataset to accurately tag all words within a discharge summary. Table 1 displays an example of part of our processed data.

Table 1. Preprocessed data

Sentence	**Word**	**Tag**
Sentence 1	patient	0
	experienced	0
	stomach	1
	pain	1
	before	0

3.5 Proposed Symptom Extraction DL Model

To perform NER we utilized Keras's biLSTM to effectively identify symptom data. From our preprocessed data frame, we utilized a rolling window technique that creates an array of tokens (e.g. *[0, 128], [96, 224]*) with length 128 and a stride size of 32 tokens (Nellis et al., 2018). This ensures that the model learns the overall context revolving around each word and prevents invaluable information from being lost. Our model consisted of an embedding layer trained using either GLoVe word embeddings or BERT sentence embeddings. In addition, we concatenated pretrained embeddings with our self-trained embeddings based on the Harvard Medical Dataset and the MIMIC-III clinical dataset to better capture the domain-specific contextual information (Zhang et al., 2021; Chang et al., 2021; Liu et al., 2020). Each biLSTM layer had a 0.2 dropout rate with output dimensions = 128, and each regular LSTM stack layer had a dropout rate of 0.5 and outputs dimensions = 128. The output then entered a single time distributed dense layer dense layer which projected a probability for each token on whether it was a symptom or not. Values closest to 1 represented a symptom and values closest to 0 represented no symptom. Figure 3 below illustrates the workflow of our deep learning model.

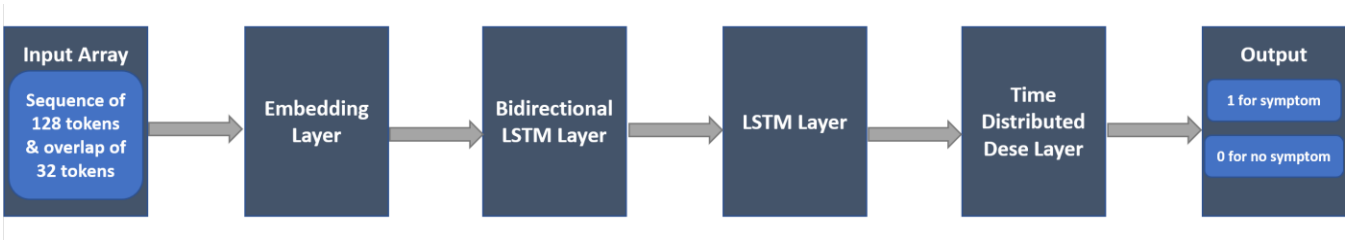


Figure 3. Basic Framework of our deep learning model

Our LSTM model was trained using a binary entropy loss function, an Adam optimizer with a learning rate of 0.001, and a stop function when validation loss ceased to improve over 2 epochs. Our test data consisted of 100 documents with high inter-annotator reliability from the MIMIC-II dataset. Using Keras's stratified K-Fold which maintains class ratio for imbalance classification, we ran 10 folds of 20 epochs using the other 909 data files with validation set equal to 0.2. We then evaluated each fold on the gold-standard set of 100 documents and compared the results. To measure the model's performance, we calculated precision, recall, and F1 score using Wilson score correction for binomial proportions.

3.6 biLSTM Experimentation

A main difficulty that arose while creating an optimal IE tool was that the dataset was extremely imbalanced. The ratio of symptom tokens to no symptom tokens was 1:42. Oversampling is the idea of duplicating the minority class (tagged) in order to create a more balanced dataset. Undersampling, on the other hand, removes certain amounts of the majority class (no tag) (Rodríguez et al., 2021; Liu et al., 2011; Chen et al., 2011). We oversampled each sentence that contained a symptom 30 times and undersampled every 5 sentences that contained no symptoms. However, these two methods lead to increased overfitting and loss of invaluable data information for our model. Sample weights negate these effects by assigning given weights to each token. We assigned each tagged token with weight 32.58 and each untagged token with weight 0.51. This enabled the model to focus on every tagged token, increasing learning and performance. Finally, after experimenting with hyperparameter batch size, we found that batch size = 128 created the most efficient runtime and achieved best results.

4. Results

4.1 Benchmarks

To develop the best performing model for symptom extraction, we created seven benchmarks utilizing different word embedding models and compared results.

GLoVe word embedding + biLSTM neural network. This pre-trained word embedding contained 300-dimensional Global Vectors (GloVe) trained on 2.2 million words computed on English Wikipedia. We then utilized this algorithm to train on our word dictionary of length 32476. This was then fed into the biLSTM neural network.

BERT-Uncased-Transformer sentence embedding + biLSTM neural network. The BERT-UncasedTransformer was pretrained on Book Corpus, a dataset consisting of 11,038 unpublished books, and English Wikipedia. This pre-trained sentence embedding transformer output dimensions 768 and trained on our input sentence tokens of length 128. This was then fed into the biLSTM neural network.

Clinical-BERT-Transformer sentence embedding + biLSTM neural network. In order to create an embedding layer more suited toward the medical domain, we experimented using the Clinical-BERTTransformer. This transformer was pre-trained on the original i2b2 medical project, creating an embedding layer that theoretically should identify greater connections between our medical words. Like the BERT-Uncased-Transformer, this transformer output dimensions 768 and trained on our input sentence tokens of length 128 and was then fed into the biLSTM neural network.

GloVe word embedding concatenated with Clinical-BERT-Transformer sentence embedding + biLSTM neural network. Using the GLoVe word embedding and Clinical-BERT-Transformer as benchmarks, we concatenated the outputs of both pre-trained models with dimension 300 in order to create an embedding layer that consisted of both word and sentence embeddings. This was then fed into the biLSTM neural network.

FT-WE embedding + biLSTM neural network. Using a pre-trained embedding model (FastText) from Facebook, we fine-tuned (changed the output layer of weights from the FastText embedding model) the FastText model on the Harvard Medical Dataset and the MIMIC-III clinical dataset. Afterwards, this embedding layer was fed into our biLSTM Neural Network.

Clinical-WE embedding + biLSTM neural network. In addition to fine-tuning, we tried self-training our own embedding model from scratch using FastText algorithms. We trained this Clinical-WE embedding on the MIMIC-III clinical dataset and the Harvard Medical Dataset. This was then fed into our biLSTM Neural Network.

Clinical-WE embedding concatenated with GloVe word embedding + biLSTM neural network. Since the GloVe embedding and the Clinical-WE embedding greatly improved performance, we decided to try concatenating the two to determine if results could be further improved. After concatenating the two layers, we fed it into our biLSTM neural network.

4.2 Metrics for symptom extraction

To accurately compare performance between each experimentation, we utilize the F1 metric to measure model performance. F1 scores are calculated on a per class basis (e.g., symptom and no symptom) and utilizes four categories to calculate performance: true positives, false positives, true negatives, false negatives. True positives is when the model accurately identifies a symptom; False positives is when the model inaccurately identifies a symptom; True negatives is when the model accurately identifies a non-symptom; False negatives is when the model inaccurately identifies a non-symptom. F1's per class calculation enables accurate evaluation for imbalanced classification tasks such as ours. Within the F1 scoring metric, there are two other metrics: Precision measures the model's accuracy on identifying true positives (symptom words) and Recall quantifies the number of correct positives made from all positive predictions (symptom words and non-symptom words). In the case of symptom extraction, it is more important that we get accurate classification of symptom words (true positives), meaning precision is more weighted than recall when calculating F1 score. Calculations are shown below.

$$Precision = \frac{True\ positives}{True\ positives + False\ positives}$$

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

4.3 Performance analysis

Table 2. Performance Comparisons

Model	F1-Score	Precision	Recall
GloVe + biLSTM	89.0	89.6	88.7
Bert-Uncased + biLSTM	78.9	82.6	76.5
Clinical-BERT + biLSTM	77.2	77.7	77.2
GloVe + Clinical-BERT + biLSTM	87.1	88.1	86.6
FT-WE + biLSTM	89.8	90.2	89.5
Clinical-WE + biLSTM	92.7	93.1	92.4
Clinical-WE + GloVe + biLSTM	**95.6**	**96.0**	**95.3**

Results for the symptom extraction comparing our pre-trained word embeddings + deep learning neural networks and fine-tuned/self-trained embeddings + deep learning neural networks are shown above in Table 2. The pre-trained GLoVe embedding concatenated with our self-trained Clincal-WE model paired with the biLSTM neural network outperformed all other models with an F1 score of 95.6, precision score of 96.0, and recall score of 95.3. On the test set, this model identified little to no false positives, indicating that this model had effectively learned to utilize surrounding context rather than simple text identification. Additionally, this model detected many symptoms outside of the training set such as (“wavy vision” or “epigastric pain”), indicating that this model was learning overall text structure and generalizing beyond the given training sets.

5. Discussion

5.1 Results and implications

Our model, the pre-trained GLoVe embedding concatenated with self-trained Clincal-WE model with the biLSTM neural network, achieves highest performance in extracting symptoms from EHR discharge summaries. Using both pre-trained and self-trained embeddings combined with biLSTM neural networks, classification performance is better than current state-of-the-art models. Unlike rule-based or dictionary-based approaches, our model extracts symptom words based on semantic and contextual information embedded in the EHR. Therefore, the model is capable of differentiating between true symptoms experienced and reported by patients versus other symptoms such as those related to medication prescription, adverse events, or health education tips documented in the EHR. Moreover, the model has generalizability compared with rule-based or dictionary-based approaches as it can recognize different terminology in describing the same symptom and new symptom words that are not included in training data. Thus, our model has the capability to distinguish symptoms in various forms including new terms and abbreviations. This is particularly important because healthcare providers use ambiguous descriptions and abbreviations to describe certain symptoms in the EHR.

Our research introduces the use of deep learning techniques within NLP to capture greater contextual elements in NER tasks. Given the increasing interest in using AI to automate information extraction from the EHR, our model has the potential to be applied across various information extraction tasks with clinical and research implications. For example, our system can be adopted in the EHR system to make symptom extraction fast, timely, and at the point of care, which should reduce EHR-burden and burnout, and enhance healthcare. In terms of research, our system could be deployed to achieve a wide range of research goals. For example, researchers interested in phenotyping patients based on their disease characteristics, including symptoms, at large scale as well as identifying patient cohorts for clinical trials could potentially use our system. Additionally, research focusing on discovering symptom patterns and clusters associated with certain diseases and treatments could benefit from our model. Moreover, our system has the potential to boost the use of big EHR data for healthcare system performance improvement purposes such as monitoring the timely documentation of certain aspects of patient care, such as

assessing and managing symptoms. Furthermore, the present work could be used to guide further research related to information extraction from the EHR using deep learning.

5.2 Limitations and future research

There are many avenues for further research as ours only introduces the surface of the potential of deep learning within IE tasks. To increase our model's performance, new deep learning techniques such as XLNet (Yang et al., 2019) could be combined with our BERT word embedding and GloVe word embedding to more accurately identify patient symptoms. In addition, this research only used the pretrained GloVe word embeddings from Google. One can fine-tune the word embedding using a large medical corpus. The fine-tuned word embeddings could help further improve the model's performance. An analysis of symptoms across languages should also be considered. Using transformer packages that utilize BERT multilingual models to create embeddings could be incorporated in future models to allow accurate symptom extraction in other languages such as Chinese and Spanish.

Our study utilized MIMIC-II dataset that only incorporated 1,009 discharge summaries. In addition, the annotation guidelines followed to identify symptoms could be considered subjective and complex, creating discrepancies within our training sets. Future IE systems should require large repositories of multi-institutional, multi-context clinical notes with high quality annotations to train on to create exact replicas of symptom extraction in the medical field. Additionally, incorporating other datasets such as the MIMIC-III medical dataset of clinical notes can be used in training sets to allow greater generalization of symptom extraction in EHRs.

Researchers could also evaluate our model on EHRs other than discharge summaries to create a usable model that covers a wide array of tasks. Despite the success in the symptom extraction task, further research should analyze methods to classify these symptoms into positive, negative, and uncertainty categories to allow for viewing symptoms in the EHR. Furthermore, a procedure for recognizing semantic similarity between similar symptoms should be developed in order to provide a "formal definition" of symptoms for professionals to use. However, this research

provides a new methodology within deep learning that could be utilized across a wide array of NER tasks in EHRs to allow healthcare providers to efficiently extract crucial medical information

6. Conclusion

This study introduces a novel approach of incorporating NLP techniques within deep learning to effectively extract symptom data from EHRs. The results from this study clearly show that deep learning is a viable option for the symptom extraction task; the utilization of novel NLP techniques with customized trained word embeddings and biLSTM neural networks significantly outperform the baselines (Steinkamp et al., 2020) and provide greater usability than traditional statistical- and lexicon-based approaches. Our proposed model achieves very promising performance results that can become the basis for improvement in future research. Additionally, our model could be used to extract other clinical information from the EHR, such as medical diagnoses, medications, and treatment plans.

References

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv*.

Li, Jianqiang, Li, J., Fu, X., Masud, M. A., & Huang, J. Z. (2016). Learning distributed word representation with multi-contextual mixed embedding. Knowledge-Based Systems, 106, 220–230. https://doi.org/10.1016/j.knosys.2016.05.045

Zhai, Z., Nguyen, D. Q., Akhondi, S., Thorne, C., Druckenbrodt, C., Cohn, T., ... Verspoor, K. (2019). Improving Chemical Named Entity Recognition in Patents with Contextualized Word Embeddings. Proceedings of the 18th BioNLP Workshop and Shared Task, 328–338. https://doi.org/10.18653/v1/w19-5035

Shih, S.-P., Jiang, J. J., Klein, G., & Wang, E. (2013, September 1). Job burnout of the Information Technology Worker: Work exhaustion, depersonalization, and personal accomplishment. Information & Management. Retrieved September 30, 2022, from https://www.sciencedirect.com/science/article/pii/S0378720613000888

Liu, N., & Kauffman, R. J. (2020, May 16). Enhancing healthcare professional and caregiving staff informedness with data analytics for chronic disease management. Information & Management. Retrieved September 30, 2022, from https://www.sciencedirect.com/science/article/pii/S0378720620302482

Feng, D., & Chen, H. (2021). A small samples training framework for deep Learning-based automatic information extraction: Case study of construction accident news reports analysis. *Advanced Engineering Informatics*, *47*, 101256. https://doi.org/https://doi.org/10.1016/j.aei.2021.101256

Hoang, T. B. N., & Mothe, J. (2018). Location extraction from tweets. *Information Processing &*

Management, *54*(2), 129-144.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holmgren, A. J., Downing, N. L., Tang, M., Sharp, C., Longhurst, C., & Huckman, R. S. (2021). Assessing the impact of the COVID-19 pandemic on clinician ambulatory electronic health record use. *Journal of the American Medical Informatics Association*. https://doi.org/10.1093/jamia/ocab268

Kalra, D. (2006). Electronic Health Record Standards. *Yearb Med Inform*, *15*(01), 136-144.

Koleck, T. A., Dreisbach, C., Bourne, P. E., & Bakken, S. (2019). Natural language processing of symptoms documented in free-text narratives of electronic health records: a systematic review. *Journal of the American Medical Informatics Association*, *26*(4), 364-379.

Mansouri, A., Affendey, L. S., & Mamat, A. (2008). Named entity recognition approaches. International Journal of Computer Science and Network Security, 8(2), 339-344.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

Mikolov, T., Chen, K., Corrado, G. s., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *Proceedings of Workshop at ICLR, 2013*.

Nellis, C., Danielson, T., Savara, A., & Hin, C. (2018). The F-t-Pj-RG method: An adjacent-rollingwindows based steady-state detection technique for application to kinetic Monte Carlo simulations. *Computer Physics Communications*, *232*, 124-138. https://doi.org/https://doi.org/10.1016/j.cpc.2018.05.013

Liu, R., Mai, F., Shan, Z., & Wu, Y. (2020, October 14). Predicting shareholder litigation on insider trading from financial text: An interpretable deep learning approach. Information & Management. Retrieved September 30, 2022, from https://www.sciencedirect.com/science/article/pii/S0378720620303256

Fan, B., Fan, W., Smith, C., & Garner, H. “Skip.” (2020). Adverse drug event detection and extraction from open data: A deep learning approach. Information Processing and Management, 57(1), 102131. https://doi.org/10.1016/j.ipm.2019.102131

Chang, Y.-C., Ku, C.-H., & Nguyen, D.-D. L. (2021, December 29). Predicting aspect-based sentiment using Deep Learning and information visualization: The impact of covid-19 on the airline industry. Information & Management. Retrieved September 30, 2022, from https://www.sciencedirect.com/science/article/pii/S0378720621001610

Zhang, M., Fan, B., Zhang, N., Wang, W. J., & Fan, W. G. (2021, Jan). Mining product innovation ideas from online reviews. Information Processing & Management, 58(1), 12, Article 102389. https://doi.org/10.1016/j.ipm.2020.102389

Olah, C. (2015). Understanding lstm networks. Colah’s blog.

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Oyelade, O. N., Obiniyi, A. A., Junaidu, S. B., & Adewuyi, S. A. (2018). Patient symptoms elicitation process for breast cancer medical expert systems: A semantic web and natural language parsing approach. *Future Computing and Informatics Journal*, *3*(1), 72-81. https://doi.org/https://doi.org/10.1016/j.fcij.2017.11.003

Patel, R. S., Bachu, R., Adikey, A., Malik, M., & Shah, M. (2018). Factors Related to Physician Burnout and Its Consequences: A Review. *Behavioral Sciences*, *8*(11), 98. https://www.mdpi.com/2076-328X/8/11/98

Robertson, S. L., Robinson, M. D., & Reid, A. (2017). Electronic Health Record Effects on Work-Life Balance and Burnout Within the I(3) Population Collaborative. *Journal of graduate medical education*, *9*(4), 479-484. https://doi.org/10.4300/JGME-D-16-00123.1

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP),

Rodríguez, N., López, D., Fernández, A., García, S., & Herrera, F. (2021). SOUL: Scala Oversampling and Undersampling Library for imbalance classification. *SoftwareX*, *15*, 100767. https://doi.org/https://doi.org/10.1016/j.softx.2021.100767

Liu, Y., Yu, X., Huang, J. X., & An, A. (2011). Combining integrated sampling with SVM ensembles for learning from imbalanced datasets. *Information Processing and Management*, *47*(4), 617–631. https://doi.org/10.1016/j.ipm.2010.11.007

Steinkamp, J. M., Bala, W., Sharma, A., & Kantrowitz, J. J. (2020). Task definition, annotated dataset, and supervised natural language processing models for symptom extraction from unstructured clinical notes. *Journal of Biomedical Informatics*, *102*, 103354. https://doi.org/https://doi.org/10.1016/j.jbi.2019.103354

Chen, E., Lin, Y., Xiong, H., Luo, Q., & Ma, H. (2011). Exploiting probabilistic topic models to improve text categorization under class imbalance. *Information Processing and Management*, *47*(2), 202–214. https://doi.org/10.1016/j.ipm.2010.07.003

Sutton, J., Ash, S., Al-Makki, A., & Kalakeche, R. (2019). A Daily Hospital Progress Note that Increases Physician Usability of the Electronic Health Record by Facilitating a Problem-Oriented Approach to the Patient and Reducing Physician Clerical Burden. *The Permanente journal*, *23*. https://doi.org/10.7812/TPP/18-221

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, *32*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 20.

Zhang, H., Boons, F., & Batista-Navarro, R. (2019). Whose story is it anyway? Automatic extraction of accounts from news articles. *Information Processing & Management*, *56*(5), 1837-1848. https://doi.org/https://doi.org/10.1016/j.ipm.2019.02.012