Predicting crashing patients with informal clinical notes by ensemble of neural document embeddings

Brenna Li
Department of Computer Science
University of Toronto
brli@cs.toronto.edu

Jienan Yao
Department of Computer Science
University of Toronto
jnyao@cs.toronto.edu

Stephen Gou
Department of Computer Science
University of Toronto
gouzhen1@cs.toronto.edu

Yuyang Liu
Department of Computer Science
University of Toronto
yuyang@cs.toronto.edu

Abstract

Clinical notes and texts data can provide more valuable and richer information than tabular data (lab results and vitals). However, the extraction and representation of such data for modelling is challenging. Less formal data such as nursing notes is even harder to model, given their vast amount of typos, acronyms and lack of grammar. We developed a method to generate effective document level embedding by ensembling two different embeddings with one embedding trained conditioned on the other. It is easy to train and flexible. We experimented on our proprietary data and showed that our method produced superior performance against baseline models in predicting whether a patient is a crashing patient with only their first 24 hour of nursing notes. In addition, we demonstrated that using abbreviation expansion significantly improves topics based document embedding such as Latent Dirichlet Allocation (LDA). We also demonstrated that pre-trained embedding on biomedical corpora does not guarantee performance improvements, given its very different language distribution from our notes data.

1 Introduction

With investment in clinical digitization we are seeing a wealth in data that can be used for predictive and analytical measures [1]. In particular, many health organizations are interested in the development of Clinical Early Warning Systems (CEW), that can assist clinicians in prioritizing treatment and care [2]. Especially in current-day hospital settings where clinicians are already understaffed and overburdened with clinical tasks, it is quite difficult to constantly keep track of a patient's state, which has negatively impacted the patient's health outcomes [3]. Therefore, the need and research in CEW systems is rising. And studies have shown that by using patient's physiological data and clinician's notes, CEW systems can assist hospital staffs in predicting at risk patients for timely treatment, and thereby reducing mortality and costs [4].

The purpose of this study is to build upon prior work in CEW systems, with a focus on the predictive power of clinical notes in anticipating whether a patient in the General Internal Medicine (GIM) ward would end up in a critical state of crashing, which is defined as death or Intensive Care Unit (ICU) transfer. More specifically, we focused on using unstructured nursing notes that are documented at the bedside during the patient's stay in the GIM ward. We took an exploratory approach by experimenting with various prepossessing, word embedding, document embedding and classification models of a

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

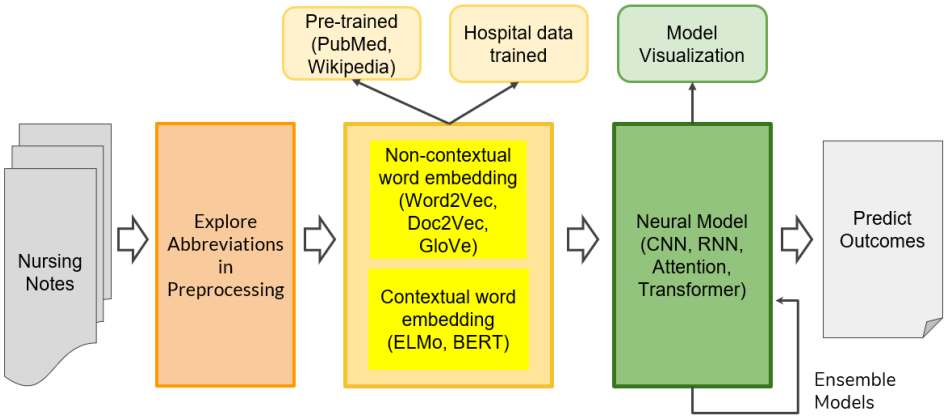

Figure 1: Overview of contributions

standard Natural Language Processing pipeline to determine which combinations yield the best result in predicting whether a patient is crashing or not.

In this paper we explore the following aspects of clinical note prediction:

1. Exploring the need and improvements to predictive ability when you apply medical abbreviation expansion on unstructured clinical notes.
2. Evaluating various document embedding techniques: doc2vec, CNN/RNN plus word2vec/GloVe, BERT, BioBERT.

Our findings and contribution from this work as summarized include:

- Demonstrating the significance in abbreviation expansion in unstructured clinical text, especially in situations with small data sample and topic based models such as LDA.
- Demonstrating that an ensemble of embeddings from two different methods can significantly outperform each individual embedding when training one embedding conditioned on the other.
- Demonstrated that pre-trained word embedding, even from the same domain, does not guarantee performance improvements when the formality of language of corpus is very different.

2 Related Work/Background

Abbreviation Expansion

Several studies have looked into the problem of ambiguous, unstructured and abbreviated text in the medical domain. Ctakes, MetaMap, CARD are among examples of work, that apply medical annotations and abbreviation look ups [5][6]. These works largely utilizes a medical dictionary, such as UMLS to disambiguate terms and evaluate accuracy. We followed a similar framework, for our abbreviation expansion task, except instead of evaluating on the accuracy of the abbreviation event, we focused on whether expansion has impact on prediction outcome, even when the expanded form may not be the correct one. To our knowledge, utilizing abbreviation task for prediction analysis is fairly unique.

Topic Models

A more tradition approach to natural language processing has been the usage of topic models. A simple example that is still popular today is the bag-of-word (BOW) approach which captures the word frequencies in a document. An improvement is the TF-IDF (term frequency inverse document frequency, which normalizes the frequency of words across all the documents. However these

approaches still have flaws. Firstly, this method can produce extremely high-dimensional vector representations that are in the range of hundreds of thousands. Secondly, it does not capture semantic relationship between words.

Latent Dirichlet Allocation (LDA) is another popular approach to generate vector representation for a document. It is a generative statistical model that assumes each document has a probability distribution over topics, and each topic has a probability distribution over vocabulary. Therefore, a document can be represented by the vector of probabilities over topics, and the number of topics is a hyper parameter for the model. LDA generates an efficient and low dimensional representation and offers decent interpretability. It produces competitive results comparing to more sophisticated methods. Given its simplicity, it is often used as a baseline model for benchmarking more advanced models.

Word Embeddings

A better way to extract semantic relationship is by using word embeddings. Word2vec, by Mikolov et al., is a popular word embedding tool [7]. It's able to generate word vectors that capture the semantic similarities such that similar words are closer together in the vector space. Word2vec can be trained through two methods: skip-gram and continuous bag of words (CBOW). In both methods the core idea is to use a neural network to map between a word and its neighboring words, or context. This forces the neural network to generate embedding that captures semantic similarities of words.

Jeffrey et al. [8] presented a matrix factorization method called GloVe (Global Vectors for Word Representation) that is a similar but faster compared to PCA. GloVe is based on the idea that words with similar distributions have similar meanings. Aimed to form a co-occurrence matrix $\mathbf{X} = V \times V$ which counts the number of times two words are appearing close to each other and V here refers to the vocabulary. GloVe approximates the $\mathbf{X} \approx \mathbf{R}\tilde{\mathbf{R}}$ where $\mathbf{R}$ and $\tilde{\mathbf{R}}$ are rank-k approximation and the cost function is the squared Frobenius norm except that entries are reweighted to only consider non-zero ones and utilize $logx_{ij}$ instead of x_{ij} in the approximated reconstruction.

Document Embeddings

CNN with 1D convolutions and RNN can be used to generate document embedding. Document is represented as a series of word embedding vectors. CNN and RNN are both able to capture contexts when scanning through the document. CNN achieves this by using filters, each aggregates a range of input values. RNN is able to store previous or future words through hidden states, which are passed along and can be preserved through the entire forward pass. Since both CNN and RNN requires fixed size input, the document can be truncated or padded to a common length, which is usually the max or average document length in the corpus. RNN is better at capturing temporally extensive dependencies in the input. However, CNN is significantly faster to train.

Doc2vec [9] is another popular document embedding, which is an extension to word2vec embedding. Doc2vec provides fixed-length dense vector representation for variable length text such as sentences, paragraphs or documents. It is obtained by concatenating paragraph vectors that are unique among paragraphs with several word vectors from a paragraph and trained under the objective of predicting the following word in the given context. Le and Mikolov [9] pointed out that the paragraph vector trained with distributed memory model represents what is missing from current context and can therefore act as a memory of the topic of the paragraph.

To our knowledge the work on double embedding from Xu, Hu et al., is closest to our idea of incorporating multiple embeddings [10]. Their idea is to combine two pre-trained word embeddings: a general-purpose embedding and a domain-specific embedding. They represent the document by concatenating two matrices, each representing the document produced by one of the word embeddings. The resulting matrix is then fed into convolutional layers and fully connected layers for classification.

There are more recent language models such as ELMo [11] and BERT [12] in which word embeddings could vary based on context, and these models pre-trained on large general domain corpus are also publicly available for downstream NLP tasks. However, these general domain models fine-tuned on our clinical text data is sub-optimal due to medical domain specific word patterns. Using models trained on biomedical corpus, such as BioBERT [13] provides some improvements, but based on our work in CNN and RNN, we theorize that the best performance would be achieved if we trained on our own nursing note. However, training our own embedding from scratch is very computationally

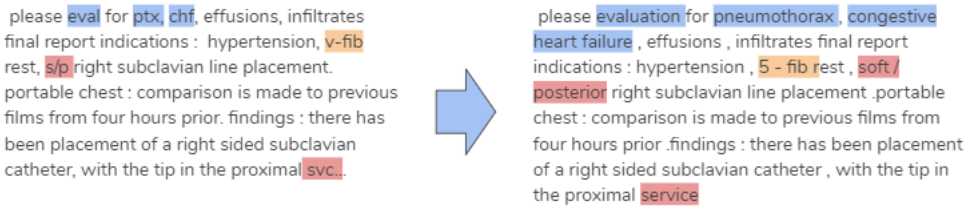

Figure 2: Example of Abbreviation Expansion on Clinical Notes

expensive and it would be necessary to consider the trade-off between more advanced models and cheaper but similarly effective models that could train embedding from our own data.

3 Method

3.1 Preprocessing

We first concatenated all nursing notes taken in the first 24 hours into one document for each patient admission encounter.

Then in the pre-processing step, we followed standard NLP procedures, including removing white-space, numerical texts and standard English stop-words using the Gensim library. In the case where abbreviation expansion was not applied we ran the pre-processing step directly, however, with abbreviation, the pre-processing came after abbreviation expansion was applied.

The Abbreviation Expansion task can largely be described as a 4 step process, abbreviation recognition, sense identification, expanded term disambiguation, and expanded term replacement in original text. This Abbreviation Expansion process is an adapted version of the Clinical Abbreviation Recognition Disambiguation (CARD) [6]. The abbreviation recognition step utilizes Support Vector Machine (SVM) to identify unique abbreviations as well as more standard abbreviation corpus such as UMLS, LRABR, and ADAM. After recognizing an abbreviated form, the algorithm determines the possible sense inventory the term can belong to. The sense inventory is constructed from the entire medical corpus by sense clustering methods. After this, is the step of disambiguating the sense of the abbreviated term to determine its expanded equivalent. The expanded form is assigned based on the sense profile and frequency of the term in the text. Finally, we replace the identified abbreviated term with its expanded version in the original text. This will then get used by following processes in our pipeline.

In Figure 2, we see an example of clinical text before and after abbreviation expansion is applied. In blue, we high-lighted, likely correct medical expansions, whereas, in orange and red, we demonstrate the likely incorrect expanded forms of the abbreviation.

3.2 Recurrent neural network

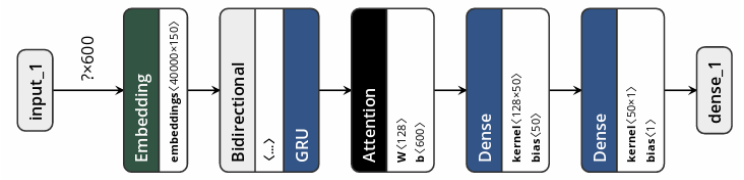

Figure 3: The architecture for the attention based RNN

The underlying ideas from aforementioned models(ELMo/BERT) are the recurrent units(LSTM/GRU) to capture temporal dynamics and the attention mechanism to help language model to attend to important words embeddings and here we present our own simplified approach to utilize these two ideas, trained on our own notes data.

Our RNN based model has its first layer as the Embedding layer with weights size **vocabulary_size** $\times$ **embeddings_dimension**, this essentially serves as a fixed look up table and its weight are not updated during training. We are relying on the recurrent hidden units to learn the dynamics and contextual relations between the words in the notes. Specifically, we concatenated the bidirectional mappings of the words, 64 hidden units in each direction, in both the forward and the backward context. We used the gated recurrent unit (GRU) as the building block for the recurrent connections to encourage simplified structure yet keeping the error signals flowing back smoothly during back propagation. Following the recurrent connections is the attention layer, which is a weight vector that behaves similarly to softmax for all the hidden units at each timestamp. The attention layer could help the recurrent units to better attend to the important words. The second last layer is a fully connected layer with 50 hidden units along with ReLU activation used as latent representation for each note that is later to be combined with the tabular data. The last layer is the sigmoid output which represents the prediction on whether the patient needs immediate care in first 24 hour of admission.

3.3 Ensembled Embedding

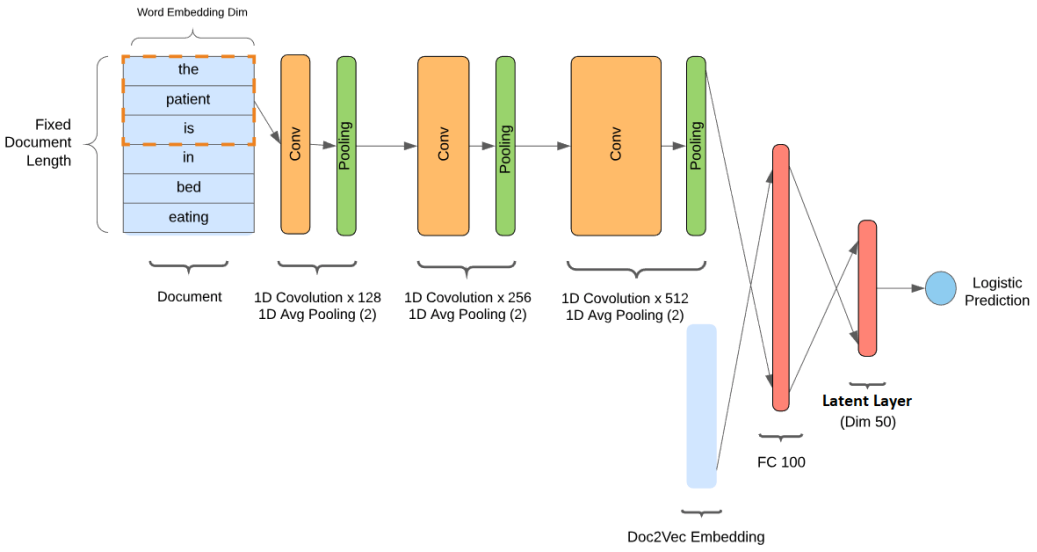

Figure 4: Model architecture for training ensembled embedding

Our model uses an ensemble of doc2vec embedding and an embedding produced by a CNN. The doc2vec model is trained first to produce a vector representation for each document. Then a CNN is trained conditioned on the doc2vec vectors, which can encourage the CNN to learn patterns that doc2Vec vectors cannot capture.

Figure 4 shows our model architecture. The input for CNN is a matrix where each row represents the embedding for a word. We truncate and/or pad the matrix to a maximum number of 300 rows. This number is determined by the average document length in our data. The word embeddings we experimented with are GloVe and word2vec. For each type of word embedding, we experimented both pre-trained and self trained. The CNN consists of three 1D convectional layers, each with double number of filters than previous one. All filters are of size 3. Each convolutional layer is followed by an average pooling layer, which we found producing better results than max pooling layer. We train a doc2vec model (PV-DM) on our nursing note dataset with 150-dimensional embeddings.

For a given piece of first 24-hour nurse note, the sequence of GloVe/word2vec embeddings is fed into CNN to extract feature vectors, then the doc2vec vector is concatenated with the output of last layer in CNN. The concatenated vector is then fed into two densely connected layers. We train the network to predict crashing patients so that the resulting embedding will be most effective and relevant in this task. We take the output of last fully connected layer (dim 50) as our latent layer that can be combined with tabular data later.

To reduce overfitting, we apply dropout (35%) layers before each fully connected layers, and we early stop training when validation error starts to increase. Parameters were optimized by informal grid search.

3.4 Model Interpretation

CNN

To help clinicians understand how the model makes decisions and to identify whether certain undesirable association were learned, we adopt a technique called grad-CAM, proposed by Selvaraju, Ramprasaath R. et al on the CNN part of our model [14]. This method was originally intended for visualizing CNN decision making on image related tasks such as image classification. It's able to produce coarse highlights over regions that generate the highest activation to the final class output. This helps identify failure models, and evaluate the model's generalizability. We adapted this method on our CNN with 1D convolution. We compute the gradients with respect to the output of the final 1D convolutional layer, which highlights the corresponding regions of the input texts that generate the most activation to the final classification. We conduct empirical evaluation over notes that are classified as positive and examine the highlighted contents. One example is provided in Figure 5. The resulting highlighted texts can be shown to clinicians to identify false association and bias.

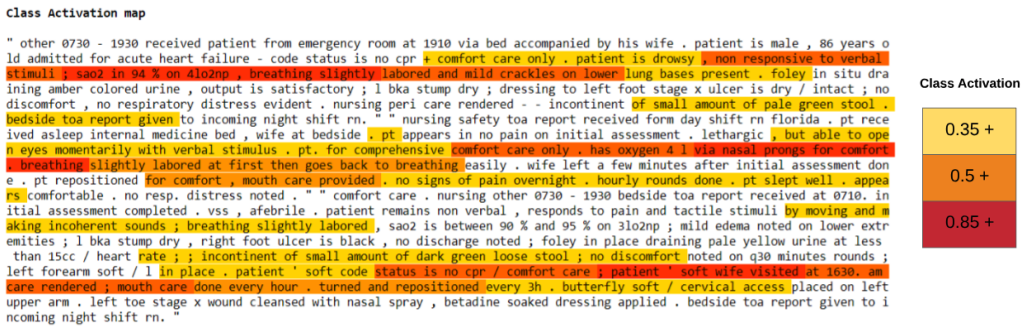

Figure 5: Example activations visualized on a piece of nursing note

Attention

A characteristic about attention is its easy-to-use visualization capability thanks to its softmax like distributions. In Figure 6 a block of preprocessed notes is augmented with the level of attention for each word within. This visualization is achieved by extracting the attention layer values from a patient classified positive and using Yang et al. code on github[15].

4 Experiment and Result

4.1 Dataset

We used de-identified clinical data from St. Michael's Hospital's General Internal Medicine department. The data consist of nursing notes and tabular physiological data that is recorded after admission. While our focus is on nursing notes, we did incorporate the physiological data, which consists of (heart rates, oxygen level, lab test results etc) for our final evaluation. The nursing notes we used is largely unstructured, free-hand and heterogeneous, with spontaneous entries. Thus only the first 24hr data aggregate was use to predict outcome, which is a binary classification of whether the patient was crashing or not. The data is separated into three groups, training, validation, and testing. Due to the fact that the data collection is subject to time, we allocated older notes, that were before 2016, to the training set which consists of 15k entries, and post 2016 data were divided between validation and testing which each contained about 1900 entries.

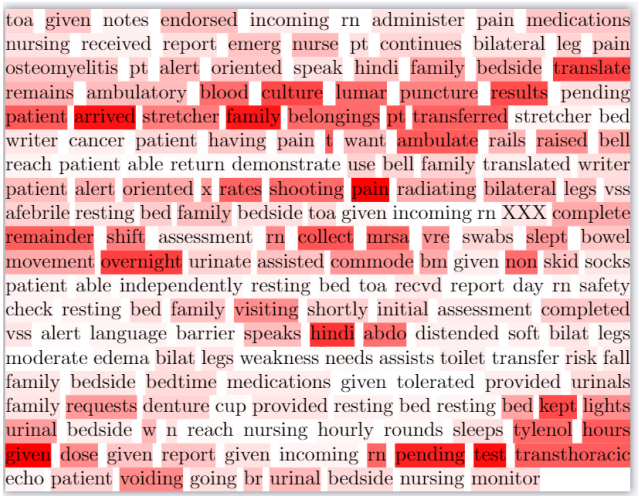

Figure 6: Example heatmap (darker means more attention) on a nursing note for a crashing patient

4.2 Baselines and evaluation

We compare our models against two baseline embeddings. The first baseline is taking the topic vector generated by a LDA model trained with number of topics equals 50. The second baseline is the vector by a Doc2Vec model. We test the performance of all embeddings by inputing the embedding into a Multilayer Perceptron for classification, and evaluate the results by AUC-ROC score. The result is shown in Table 1.

We also evaluate our top methods combined with the the physiological data (tabular) by concatenating the document embedding with the physiological features. The combined features is then fed into a Multiplayer Perceptron for classification. The result is shown in Table 2.

Table 1: AUC-ROC of models on test dataset

Models	**Standard Preprocessing**	**w Abbreviation Expansion**
LDA	0.701	0.738
Doc2Vec	0.711	0.710
CNN + Pub Med W2V	0.542	0.564
CNN + W2V	0.740	0.745
CNN + Glove	0.725	0.723
RNN + Glove	0.719	0.739
BERT	0.714	0.720
BioBERT	0.741	0.741
Embedding Ensemble	0.789	**0.791**

Table 2: AUC-ROC of models combined with tabular data

Models	**AUC-ROC**
Tabular Only	0.766
LDA	0.778
Doc2Vec	0.795
CNN + W2V	0.813
RNN + GloVe	0.811
Embedding Ensemble	**0.822**

4.3 Result

We have the following findings from the result of our experiments on different models.

- Our ensembled embedding model achieves the best result over the other models utilizing a single type of embedding. This suggests that our training technique that trains the second embedding conditioned on the first embedding is forcing the second embedding to learn patterns that's not captured by the first embedding, thus improving overall quality of final embedding.
- Abbreviation expansion greatly improves the quality of LDA topic vectors, but marginally improves neural embeddings. This could be that neural embeddings are able to capture semantic similarities, thus abbreviation are mapped to similar vector space locations as their expanded words.
- Pre-trained word vectors from PubMed perform poorly. This suggests that even though the vectors were trained on the same domain, the difference in form of language and structure greatly reduces the effectiveness of pre-trained vectors. This might be due to the fact that PubMed trained on corpus consists of medical journals and books, but our data is unstructured nursing notes, containing large amounts of typos, acronyms and grammatical errors.
- BERT also suffers from transferring embeddings learned on general domain corpus to our own tasks. An improvement can be seen when BioBERT pretrained model was employed but it performed worse than our simpler neural models.
- All embeddings combined with physiological data perform better than tabular data alone. This suggests that unstructured nursing notes contain useful information that is not in physiological data in determining crashing patients.
- CNN and RNN based latent layer produce similar results. Also, GloVe and word2vec word embeddings produce similar results in the final latent layer. However, RNN takes much longer training time.

5 Discussions

We developed a method to effectively leverage informal clinical notes for identifying crashing patients. Our contribution is two folds. 1) We showed that abbreviation expansion (CARD more specifically) can improve quality of topic representation of documents like LDA significantly, and improve neural embeddings slightly. 2) We found that ensemble of embeddings from two different methods significantly outperformed each individual embedding.

In future works, we plan to investigate the value in our prediction outcomes. As mentioned by one of our clinical advisers, in our current model, it is difficult to know whether the model is useful in predicting unforeseen patient crashes or foreseeable predictions clinicians had already anticipated.

6 Limitations

We trained our ensembled embedding by predicting crashing patients, thus the resulting embedding is optimized for this particular task. We have not tested how the embedding would generalize for other tasks.

Our method cannot capture numerical values and their significance, since we removed all pure numerical values from the clinical notes. Thus, our models would not associate an older patient with higher risk or associate abnormal vital signs with risks. Incorporating numerical values in language modelling requires further research, simply generating word embedding vectors for numbers are not realistic given the unbounded number of numerical values.

As with the case of other word embedding pre-trained on external text corpus, recent advanced models like BERT might need to be trained on our own data from scratch before we could fully utilize its enhanced modeling power. Future direction may include training and fine-tuning these models with increased amount of computational resources due to their inherently high cost of pre-training.

We could also apply the transformer architecture (pure attention) to remove the temporal dependency inherited from RNN to speed up the training process.

References

[1] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew L. Beam, and Rajesh Ranganath. Opportunities in Machine Learning for Healthcare. *arXiv:1806.00388 [cs, stat]*, June 2018. arXiv: 1806.00388.

[2] Ben Wellner, Joan Grand, Elizabeth Canzone, Matt Coarr, Patrick W Brady, Jeffrey Simmons, Eric Kirkendall, Nathan Dean, Monica Kleinman, and Peter Sylvester. Predicting Unplanned Transfers to the Intensive Care Unit: A Machine Learning Approach Leveraging Diverse Clinical Elements. *JMIR Medical Informatics*, 5(4), November 2017.

[3] Peter McQuillan, Sally Pilkington, Alison Allan, Bruce Taylor, Alasdair Short, Giles Morgan, Mick Nielsen, David Barrett, and Gary Smith. Confidential inquiry into quality of care before admission to intensive care. *BMJ*, 316(7148):1853–1858, 1998.

[4] Li-wei Lehman, Mohammed Saeed, William Long, Joon Lee, and Roger Mark. Risk Stratification of ICU Patients Using Topic Models Inferred from Unstructured Progress Notes. *AMIA Annual Symposium Proceedings*, 2012:505–511, November 2012.

[5] Ruth Reátegui and Sylvie Ratté. Comparison of MetaMap and cTAKES for entity extraction in clinical notes. *BMC Medical Informatics and Decision Making*, 18(Suppl 3), September 2018.

[6] Yonghui Wu, Joshua C. Denny, S. Trent Rosenbloom, Randolph A. Miller, Dario A. Giuse, Lulu Wang, Carmelo Blanquicett, Ergin Soysal, Jun Xu, and Hua Xu. A long journey to short abbreviations: developing an open-source framework for clinical abbreviation recognition and disambiguation (CARD). *Journal of the American Medical Informatics Association*, 24(e1):e79–e86, April 2017.

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[9] Le Quoc and Mikolov Tomas. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.

[10] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. Double embeddings and cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1805.04601*, 2018.

[11] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: pre-trained biomedical language representation model for biomedical text mining. *arXiv preprint arXiv:1901.08746*, 2019.

[14] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[15] Jie Yang and Yue Zhang. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.