Predicting Noteworthy Utterances From Doctor-Patient Conversations

Joachim H. Talloen
Tepper School of Business
and Department of Social and Decision Sciences
Carnegie Mellon University
Pittsburgh, PA 15213
jtalloen@cmu.edu

Youna Song
Department of Statistics and Data Science
Carnegie Mellon University
Pittsburgh, PA 15213
younas@andrew.cmu.edu

Chris Lee
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
chunglee@andrew.cmu.edu

David Chao
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
dchao1@andrew.cmu.edu

Abstract

Physician burnout is an important problem in the medical profession. Administrative tasks have been linked to physician burnout with physician Electronic Health Record keeping being one of the major administrative tasks physicians engage in. In the existing literature, researchers have leveraged doctor-patient conversations to predict patient diagnosis, symptoms, and medication regiment in an effort to alleviate some of the burden of note taking for physicians. Using the unprocessed doctor-patient conversations repeatedly leads to worse model performance than using a subset of the conversations. In this paper we use state of the art classification models to predict noteworthy sentences from doctor-patient conversations. We find the most significant improvements when adding context to a RCNN model as well as fine-tuning a BERT pre-trained model. We were able to achieve a 3% increase from our 69% logistic regression baseline accuracy for this binary classification task. We believe we have laid a foundation for future research trying to predict noteworthy utterances from unfiltered doctor-patient conversations and hope further improvements can be made upon our findings so far.

Introduction

A serious issue in today's medical profession is physician burnout. One of the main tasks a physician works on during a day is record keeping of patient diagnosis, illnesses or general prognosis. In particular, physicians have to enter patient relevant information into an Electronic Health Record (EHR) for each patient visit; for each half an hour patient visit a physician spends about 22 minutes entering the relevant patient information into the EHR [12]. The significant time doctors spend on administrative tasks such as EHR keeping is positively correlated with physician burnout [3, 8] and in the United States 50% of physicians report experiencing burnout at their job [6]. The extensive EHR keeping could be a main driver of physician burnout and alleviating physician note taking could significantly decrease physician burnout and improve physician mental heath.

In the recent literature there has been a surge of interest in trying to alleviate this note-taking process from physicians by using doctor-patient conversations to predict and extract EHR relevant information such as patient diagnosis [5], medication regiment (i.e. the medication a patient has to take and how frequently) [11], and symptoms [10]. A repeated challenge in extracting important

(count) Sentences	(count) Noteworthy Sentences
(1) “I take it at night.” (2) “Okay.” (3) “And I take 2 tablets and I think it’s better because I used to have to take 4 tablets.” (4) “Okay.” (5) “So I’m down to 2 and it, it’s, I’m hoping that it’s making my diabetes do better.” (6) “I hope so too.”	(1) “I take it at night.” (2) “And I take 2 tablets and I think it’s better because I used to have to take 4 tablets.”

Table 1: Sample Dialogue Illustrating Redundancy in Conversation

information from doctor-patient conversations is the data. In particular, not only is the data hard to come by, but moreover, even when the data is accessible, doctor-patient conversations are long and noisy. Each conversation has around 1500 words and often contains redundancies, informal conversation as well as complex technical jargon (see Table 1). Using this data directly as input to an encoder [11] or a BERT neural net [5] has been shown to yield worse performance than using a subset of the doctor-patient conversations when trying to predict patient diagnosis, medication regiment or symptoms. In the extant literature, a lot of focus has been on extracting useful information from doctor-patient conversations such as medication regiment as well as diagnoses, among others. However, currently no research exists on trying to extract important subsets of the entire doctor-patient conversations to be used as input into the main diagnosis or symptom prediction model. Using a subset of already extracted noteworthy sentences from doctor-patient conversations has been shown to improve these model's performances. In this paper we focus on extracting noteworthy sentences from unfiltered doctor-patient conversations, a crucial step lying in between using the raw doctor-patient conversations to make filtered physician EHR notes.

The dataset we use for this task is a corpus of 6,693 real doctor-patient conversations recorded by patients using Abridge[1]. In addition, the dataset is annotated by health experts and important (or, “noteworthy”) sentences have been extracted. We formulate the problem of predicting these noteworthy sentences as a binary classification problem and use simple learning models as well as a battery of state of the art task classification models to predict noteworthy sentences. In particular, as a first baseline, we use a logistic regression akin to that used by Krishna et al. [5] and replicate their results. In addition to implementing a neural net baseline, we also implement a CNN, RCNN, as well as a pre-trained and fine-tuned BERT model. We find the best performance with our pre-trained BERT models, achieving a 3% increase in accuracy over the logistic regression baseline.

Dataset

Our dataset consists of de-identified transcripts of real doctor-patient conversations accompanied by medical-expert notes that annotate the important sections of the doctor-patient conversation.

Each conversation has an average length of 9m 28s. The average word count of a conversation is 1500 words, and there are around 200-250 utterances in each conversation. The conversations are inherently noisy due to the noisiness of natural conversation. Specifically, there are a lot of redundant phrases such as “uhmm” and “okay” (see Table 1), cut off sentences, and small talk. There are a total of 6,693 conversations including 2732 cardiologist visits, 2731 family medicine visits, 989 interventional cardiologist visits, and 410 internist visits. The data has been de-identified by replacing personal information with digital zeros and [de-identified] tags.

Every conversation, or transcript, has been professionally annotated by a medical expert. The notes taken by medical experts follow a SOAP note format and are broken down into the following categories: Subjective, Objective, Assessment and Plan. The SOAP notes mimic the usual notes a medical expert would make (e.g. write a short description of the diagnosis) and annotate the

[1] We are grateful to Abridge for providing us with the annotated doctor-patient conversations.

transcript with where the evidence for the notes came from. For example, the Objective category records events such as lab tests that may have been performed on the patient. If bloodwork had been performed on the patient, the Objective category would include a “Bloodwork” tag with a timestamp corresponding to the respective tag. Further, it would also include the free form note a doctor would make: “Laboratory workup”, with a list of timestamps that contributed to the complete note.

We will predict two different types of noteworthy utterances: All-Noteworthy (AN) and Diagnosis-Noteworthy (DN) utterances.

- All-Noteworthy utterances (AN): utterances used as evidence in the SOAP notes
- Diagnosis-Noteworthy utterances (DN): utterances used as evidence for an entry in the SOAP notes that contains the ground truth for the diagnosis of the patient

By using both an indiscriminate and a tailored approach to selecting the data, we are able to gain more insight into how well the model can generalize. Our final dataset thus contains a list of utterances (or sentences) in one column, and whether it was noteworthy or not indicated by a 1 or 0, respectively, in the next column.

Methods

Text classification is the task of categorizing a text document into a set of pre-defined categories based on its keywords or content. In NLP and related fields, text classification has been widely used to aid in tasks like topic labeling and sentiment analysis. We find that our problem of predicting noteworthy sentences from our dataset of doctor-patient conversations lends itself to a text classification approach in which the goal is to classify sentences as either being “noteworthy” or “not noteworthy”. In what follows, we first describe the two baseline models we implemented. Thereafter, we delve into more sophisticated approaches we used such as an RNN, RCNN and a pre-trained BERT model.

Baselines

For all the baseline datasets, we computed the term frequency-inverse document frequency (TF-IDF). In particular, we compute the TD-IDF scores for each sentence with respect to all other sentences in the dataset, respectively, for each the train, validation and test dataset. Thus, both the logistic regression (LR) and neural net (NNet) baselines are run using the TD-IDF training data representations to classify the sentence importance.

We re-implemented the method for automated noteworthy utterance retrieval using logistic regression as was used in Krishna et al. [5]. In particular, we use exactly the same parameters as Krishna et al. [5]: a limited-memory Broyden–Fletcher–Goldfarb–Shanno (lbfgs) solver with 250 epochs and no L2 regularization.

We also implemented a baseline multi-layer neural net with the the following dimensions for it’s hidden layers: (128, 256, 512, 1024, 1024, 1024, 512, 256). Each layer used ReLU activation functions; an initial starting learning rate of 1e-3 was used with no learning rate scheduler. We used Adam to optimize the model, with L2 regularization weight of 1e-4, and first and second momentum decay rate weights of 0.9 and 0.999 respectively. We use no other regularization such as batch normalization or dropout and run the model for a total of 10 epochs.

RNN

Our first serious model was a simple Recurrent Neural Network (RNN) model. Instead of using the TF-IDF embeddings as inputs to our model, we use a pre-trained GloVe dictionary of size 50. The model consists of an embedding layer of size 50 followed by three bidirectional Long Short-Term Memory (LSTM) layers of size 256 each. We pad and pack our sequences before feeding them into our LSTMs and finally feed into two linear layers. The first linear layer is of size (512, 512) and the final linear layer which outputs the logits is of size (512, 2).

To optimize our model we utilize an Adam optimizer and minimize cross entropy loss. The Adam optimizer is set to an initial learning rate of 1e-3 with weight decay of 1e-5. We do not use a learning rate scheduler and train for a total of 20 epochs.

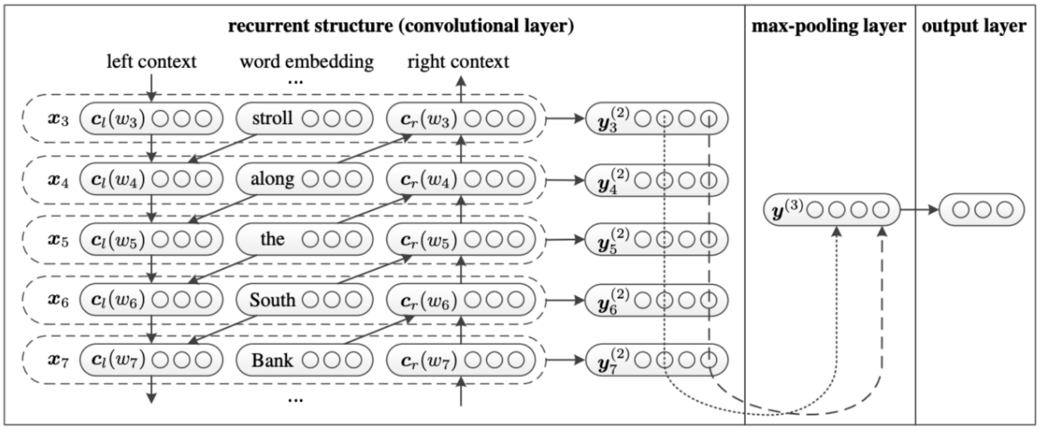
recurrent structure (convolutional layer)
max-pooling layer
output layer
left context
word embedding
right context
...
x_3 $c_l(w_3)$ stroll $c_r(w_3)$ $y_3^{(2)}$
x_4 $c_l(w_4)$ along $c_r(w_4)$ $y_4^{(2)}$
x_5 $c_l(w_5)$ the $c_r(w_5)$ $y_5^{(2)}$
x_6 $c_l(w_6)$ South $c_r(w_6)$ $y_6^{(2)}$
x_7 $c_l(w_7)$ Bank $c_r(w_7)$ $y_7^{(2)}$
...
$y^{(3)}$

Figure 1: RCNN from Lai et al. [7]

RCNN

We implement a successful approach to text classification that was developed by Lai et al. [7] in which they implement a Recurrent Convolutional Neural Network (RCNN). A Recurrent Neural Network (RNN) is useful for being able to store information from previous input text in a context layer. At the same time, a Convolutional Neural Network (CNN) may be able to better learn the semantics of the text but is not usually suitable for context-sensitive data. Lai et al. find a hybrid of the two to be successful given the nature of the text classification task as the hybrid model allows to both preserve context and identify the key sub-features of our text. In particular, first the authors utilize a bi-directional recurrent structure to capture both past and future information for learning the text representations. Thereafter, the authors feedforward through a CNN max-pooling layer. Because we suspect that particular words may be more indicative of a noteworthy classification in our task, the convolutional layer may be particularly useful in identifying the key words.

We apply this model to our current task of predicting noteworthy utterances. In particular, we utilize three bidirectional-LSTM layers of size 256. The output from the LSTMs are concatenated with the original input into a linear layer of size 256 and fed into a Tanh activation layer. Finally, the result is passed through a max-pooling layer and a final linear layer to classify it into two output classes (0 = not-noteworthy, 1 = noteworthy). During training, we utilize an Adam optimizer to minimize cross entropy loss.

We improve upon the RCNN from Lai et al. by incorporating sentence level context into our model. In particular, we found that neighboring sentences were naturally highly related to one another and including information from the context of an input sentence may help to increase accuracy. Thus, we concatenated c number of sentences from before and after a particular input as the input to our model, where c was a hyper-parameter we tuned. In our final RCNN model, we ended up using the value $c = 2$.

In addition, we applied pre-trained GloVe word embeddings of size 300 to learn more complex vector representations of the input sentences. We also added regularization techniques including locked dropout between bi-LSTM layers and embedding dropout as well as a learning rate scheduler initialized to 1e-3.

BERT

We also utilized Bidirectional Encoder Representations from Transformers (BERT), which is a pre-trained network from Google that uses the transformer architecture [4]. The transformer architecture is able to avoid the sequential nature of RNNs and process the attention of the input sequence at once. It does so by computing multiple attention values based on different metrics, a mechanism known as multi-head attention. The attention values are calculated through matrix multiplications and can be computed simultaneously, making transformers far more suited for parallel computation than traditional mechanisms of processing language like RNNs. BERT has been pre-trained on a large

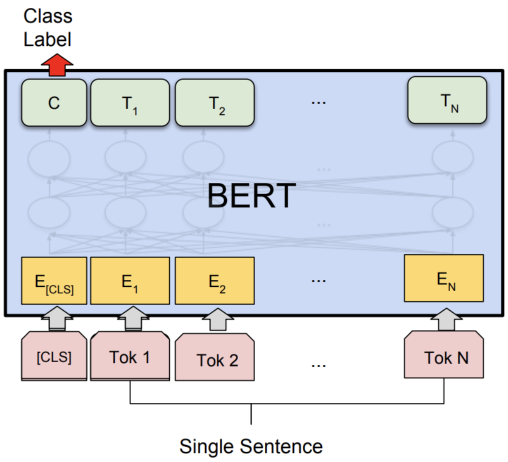

Figure 2: BERT Single Sentence Classification [4]

number of examples, giving it the ability to represent more intricate relationships within a language. As a result, fine-tuning BERT has been successful in many language understanding tasks. By adding a linear classifying layer on top of BERT, it can be used for a variety of different classification tasks. We used a module called "BERT for Sequence Classification" from HuggingFace, where we were able to designate two desired labels, noteworthy and not-noteworthy, and fine tune the model for our data. We used "Clinical BERT", which is a BERT network that had been trained specifically for the medical industry using clinical terms [1]. We trained the network for 1 epoch with batch size 32 and maximum sequence lengths 128 and 200.

We improve upon single sentence classification with BERT by incorporating sentence level context. By feeding in neighboring sentences, BERT is better able to understand the context from the dialogue surrounding the input sentence, resulting in greater accuracy. Like in the RCNN, we concatenated c sentences before and after the particular input sentence, and in our final model we used $c = 1$ for BERT.

We initially investigated applying a LSTM network on top of BERT, using the encoding values from BERT as inputs to the LSTM. We quickly discovered that BERT was not intended to be used as a simple embedding layer, and was powerful enough to be a standalone network. The performance results from the added LSTM network were poor, and we quickly transitioned into fine tuning the BERT for Sequence Classification network.

Results

Metrics

We use three main metrics to evaluate the performance of our models: accuracy, F1 score, and area under the receiver-operator characteristics (AUC).

Baselines

The logistic regression baseline scores are given in Table 2. We were able to closely replicate the results found in Krishna et al. [5]. In addition, we found that a simple MLP baseline performed slightly worse compared to our logistic regression baseline.

RNN/RCNN

On the AN dataset, we were able to see a slight improvement over our MLP baseline using a simple RNN. By adapting the RCNN model from Lai et al. [7] as well as scaling up our GloVe word embeddings, we were able to see additional improvements over the simple RNN achieving an accuracy of 0.7032. The most significant improvements in our model performance were found when we added sentence level context. Specifically, when we added two sentences before and after our current sentence, we saw a more than 1 percentage point improvement over our RCNN without context, achieving an accuracy of 0.7159.

BERT

In our BERT models we see a similar story; adding sentence level context was crucial to increasing our model performance and we were able to achieve our best results with BERT with sentence level context. For classifying the AN dataset, our best accuracy of 0.7242 came with BERT with maximum sequence length of 200. For classifying the DN dataset, our best accuracy of 0.8303 came with BERT with maximum sequence length of 128. We investigated longer maximum sequence lengths and smaller batch sizes but this proved to be too memory-intensive. We were able to achieve the highest F1 scores using BERT as well, with 0.5941 for AN and 0.2722 for DN on BERT with a maximum sequence length of 128. We were not able to surpass the AUC score of 0.6917 achieved by the Logistic Regression baseline with any of our models, but BERT came the closest with 0.6836.

Model	Accuracy	F1	AUC
AN:			
Logistic Regression Baseline	0.6990	0.4486	**0.6917**
NNet Baseline	0.6764	0.4317	0.6302
RNN	0.6976	0.4795	0.6314
RCNN	0.7032	0.4516	0.6224
RCNN w/ context	0.7159	0.5056	0.6412
BERT	0.6999	0.5057	0.6411
BERT w/ context & max seq len 128	0.7057	**0.5941**	0.6836
BERT w/ context & max seq len 200	**0.7242**	0.5342	0.6615
DN:			
Logistic Regression Baseline	0.8183	0.1284	**0.6600**
NNet Baseline	0.7828	0.2481	0.5748
BERT w/ context & max seq len 128	**0.8303**	**0.2722**	0.5750
BERT w/ context & max seq len 200	0.8299	0.2171	0.5570

Table 2: Results of Models. AN: predicted all noteworthy sentences. DN: predicted diagnosis noteworthy sentences.

Discussion

The results present two insights that were crucial for getting our best performances: (1) pre-trained BERT and (2) sentence level context.

Pre-trained BERT models have repeatedly been shown to improve model performance and the same holds true in our medical text classification task. Our best BERT models outperformed our RCNN by about 1 percentage point in the AN dataset, and these strong results generalize to the DN dataset. It is worth noting however that for the larger AN dataset having a sequence length of 200 improved performance significantly compared to a sequence length of 128, while for the smaller DN dataset both a sequence length of 128 and 200 perform comparably. Thus, having a larger sequence length seems to improve generalizability. That being said, although we achieved our best results with the BERT model, we felt that there was still room for improvement. With our implementation, increasing our maximum sequence length (to more than 200) to account for larger sentences and contexts would likely provide improvements similar to the improvement seen by increasing it from 128 to 200. As discussed in Sun [2], BERT fine-tuning for text classification can be improved through more pre-training and fine-tuning strategies. Performing some further pre-training of ClinicalBERT on the text corpora at hand, which is unique in that it is comprised solely by dialogue, would be a good next step.

Our results highlight the importance of adding context. In the RCNN with context we found that the optimal performance was achieved when we added 2 sentences before and after our current sentence. However, in BERT we found that only 1 sentence on each side resulted in the best performance. Thus, overall, while sentence level context was very important, the optimal amount of context varies across implementations.

In future research, researchers should strongly consider using pre-trained models. In particular other pre-trained models such as ELMo [9] present a natural next step. In addition, future work should aim to train the feature extraction and text classifications tasks separately as well as explore potential novel approaches to this problem such as applying GANs.

Conclusion

In conclusion, we tried a lot of different approaches given the two month time horizon for this project: starting all the way from the beginning in trying to get our hands on this unique dataset to running a series of fine-tuning experiments using pre-trained BERT models. That being said, given our task of binary text classification, achieving an accuracy of about 72% with a pre-trained model such as BERT is low. We attribute the difficulty of increasing this score mainly to the inherent noisiness of doctor-patient conversations. In particular, doctor-patient conversations are riddled with non-patient examining small talk which is dispersed at random intervals. We hope this project has laid the foundation for future research trying to help alleviate physician burnout by predicting noteworthy utterances from doctor-patient conversations. Future research may take from the insights gained in this project; concretely that both using a pre-trained model and adding context are important to improving model performance.

References

[1] E. Alsentzer, J. Murphy, W. Boag, W. Weng, D. Jin, T. Naumann, and M. McDermottt. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop, Association for Computational Linguistics*, January 2019.

[2] Sun C., Xipeng Q., Yige X., and Xuanjing H. How to fine-tune bert for text classification?, 2019.

[3] M.G. del Carmen, J. Herman, S. Rao, M.K. Hidrue, D. Ting, S.R. Lehrhoff, S. Lenz, J. Heffernan, and T.G. Ferris. Trends and factors associated with physician burnout at a multispecialty academic faculty practice organization. *JAMA open network*, 2(3):e190554–e190554, 2019.

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

[5] K. Krishna, A. Pavel, B. Schloss, J. P. Bigham, and Z. C. Lipton. Extracting structured data from physician-patient conversations by predicting noteworthy utterances. In *Working Paper*, January 2020.

[6] S. Kumar. Burnout and doctors: prevalence, prevention and intervention. *Healthcare*, 2(3):37, 2016.

[7] S. Lai, L. Xu, , K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *AAAI Conference on Artificial Intelligence*, 2015.

[8] R. Leventhal. Physician burnout addressed: How one medical group is (virtually) progressing. *Healthcare Innovation*, 2018.

[9] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[10] A. Rajkomar, A. Kannan, K. Chen, L. Vardoulakis, K. Chou, C. Cui, and J. Dean. Automatically charting symptoms from patient-physician conversations using machine learning. *JAMA Internal Medicine*, 2019.

[11] S. P. Selvaraj and S. Konam. Medication regimen extraction from medical conversations. In *Proceedings of International Workshop on Health Intelligence (W3PHIAI) of the 34th AAAI Conference on Artificial Intelligence*, January 2020.

[12] C. Sinsky, L. Colligan, L. Li, M. Prgomet, S. Reynolds, L. Goeders, J. Westbrook, M. Tutty, and G Blike. Allocation of physician time in ambulatory practice: a time and motion study in 4 specialties. *Annals of Internal Medicine*, 2016.