Simple Transformers for PHI De-identification

Aditya Khandelwal
Department of Computer Science
Stanford University
akhand@stanford.edu

Arjun Soin
Department of Computer Science
Stanford University
asoin@stanford.edu

Abstract

Data anonymization is a crucial prerequisite to clinical data sharing, transparency and follow up scientific analyses. Any such data shared must necessarily protect Personally Identifiable Information (PII). Natural Language Processing (NLP) techniques have been found useful for anonymization of patient notes prescribed by doctors, effectively removing Personal Health Identifiers (PHIs) from Electronic Health Records (EHRs) found in medical datasets. Our goal is to identify and extract named entities within discharge summaries by utilizing a novel NLP architecture, namely Transformer. Notably, we train a Transformer model on a large corpus of patient notes. This novel method is aimed at reducing the time taken to train the deep learning model whilst also achieving near-state-of-the-art results during prediction.

Add a couple sentences about results

1 Introduction

Data anonymization of clinical and patient-level data is salient to ensure transparency while sharing patient information. To protect patient confidentiality, notes accessed by medical investigators and other stakeholders must be de-identified. In the United States, the Health Insurance Portability and Accountability Act (HIPAA) defines 18 types of protected health information (PHI) that needs to be removed to de-identify patient notes. Manual de-identification, though widely utilized, can become impractical given the size of EHR databases, associated costs and human errors involved. Henceforth, a reliable and automated alternative would be invaluable. Automated de-identification systems can be classified into two categories: rule-based systems and machine-learning-based systems. Rule-based systems typically rely on patterns, expressed as regular expressions and gazetteers, defined and tuned by humans, though are not robust to language changes (e.g., variations in word forms, typographical errors, or infrequently used abbreviations) and cannot easily take into account the context.

Latest developments in the intersection of deep learning and NLP have been found useful for such automation and to overcome shortcomings of rule-based systems. Specifically, artificial neural networks (ANNs) have been employed to achieve state-of-the-art results on de-identification of datasets for patient notes [1]. The Bidirectional long short-term memory networks (BiLSTM) have been widely used in models solving the named entity recognition (NER) task, and consequently in patient de-identification. Such approaches do not require handcrafted rules or features, and can automatically learn effective features by performing composition over token embeddings. Patient information de-identification is treated as a task similar to that of named-entity recognition, where words in a sentence are identified based on pre-defined tags. In this context, the tags would take the form of personal health indicators, such as name, age, date or location, to name a few. Despite hugely encouraging results, such systems compromise on consistent and accurate de-identification of sensitive categories in the absence of gazetteer features based on the local institution's patient and staff census. Equally important to note is their lack of promise in scaling from the point of view of

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

deploying off-the shelf de-identification systems, due to limitations on parallelization at training time that stem from maintaining and updating a hidden state within the architecture, along with sequential dependencies between outputs.

Given the aforementioned context, we aim to experiment with improved performance and using novel a architecture that could ultimately lead to an efficient and scalable PHI anonymization system. A new generation of NLP models known as Transformer architectures can learn long-term dependencies from textual information. A transformer is an encoder-decoder architecture model which uses attention mechanisms to forward a more complete picture of the whole sequence to the decoder at once rather than sequentially. The biggest benefit, however, comes from how the Transformer lends itself to parallelization and significantly reduced training times. While originally the performance of Transformer on NER was unable to match its performance on other NLP tasks, recent model adaptations that incorporate direction, relative distance aware attention and the un-scaled attention, have yielded a Transformer-like encoder to be just as effective for NER as other NLP tasks [4]. In fact, a Transformer model has also been utilized for a biomedical NER task, lending credence to its usage in the health domain [5].

At the outset, we work with an existing ANN-BiLSTM model lay out the grounds for comparisons with the eventual Transformer approach. We then present a variant of the Transformer-based model known as BERT inspired by the context above where PHI labels as tags are analogous to entities in a conventional NER task.

2 Related Works

2.1 ANNs and Related Models

A first of its kind de-identification system based on ANNs was recently published [1] with inspiration from the promising performance of ANNs for various NLP tasks, such as language modeling, text classification and question-answering. The underlying model is built as a bi-directional entity LSTM and is supplemented by a program **NeuroNER** developed by Dernoncourt et al. at MIT [2]. The model serves as state-of-the-art for NER on medical datasets and contains libraries to suitably transform data into appropriate medical formats - facilitating ease of hyperparameter tuning and integration with TensorBoard. Here the authors clearly outline using quantitative evaluations that ANN models better incorporate context and are more flexible to variations inherent in human languages.

Also in the fray is a hybrid system that generated the most effective results for *The 2016 Centers of Excellence in Genomic Science (CEGS) Neuropsychiatric Genome-scale and RDOC Individualized Domains (N-GRID) clinical natural language processing (NLP) challenge*. The hybrid system is developed from four individual subsystems, combining variants of BiLSTEM and conditional random field (CRF), the results of which are merged with a rule-based system [10]. While achieving performance (F1 scores) better than those seen in state-of-the-art systems, this is a "best of both worlds" approach, which again does not necessarily scale effectively to off the shelf deployment. However, it notably confirms the viability of using Recurrent Neural Networks (RNNs) in the form of BiLSTMs to identify information from data.

2.2 Transformers

We refer readers to the original papers for an exhaustive background description for both Transformer [6] and BERT [9]. The task of biomedical named entity recognition, relevant to disease and chemical extraction has been undertaken [5] and found to outperform previous state-of-the-art systems for slot tagging on the different benchmark biomedical datasets in terms of (time and memory) efficiency and effectiveness. Another salient example of utilizing transformers in the medical domain comes from BEHERT, a deep neural sequence transduction model for EHR (electronic health records), capable of multitask prediction and disease trajectory mapping [7]. Otherwise, there has been little work around the use of transformers on PHI de-identification. All such work in the medical domain concerning token classification has crucially highlighted the importance of constructing a high quality, consensual gold set, a learning that informs our approach towards this research.

3 Approaches

3.1 Baseline: Bi-Directional LSTM Model Tuning

For this section, the goal was to reproduce the **NeuroNER** [2] model from "De-identification of Patient Notes with Recurrent Neural Networks" [1] and aim to improve it by implementing Random Search to tune hyperparameters.

The baseline model is based on a conditional random field (CRF), and is found in Section 2.1 of the aforementioned paper [1]. The model we ran from **NeuroNER** [2] is implemented as a type of Recurrent Neural Network (RNN) kown as Long Short Term Memory (LSTM) with layers for character-enhanced token embeddings, label predictions and label sequence optimization. We build on top of the given model to implement random grid search that evaluates the available range of hyperparameters for the ANN model. The grid search goes through a list of possible configurations and outputs the **NeuroNER** [2] best model with an associated set of hyperparamters.

We write a script that runs repeated experiments of the **NeuroNER** model using *random* and *subprocess* modules in python, which allow us to execute the random search process and sample multiple experiments. The chosen space for hyperparameter random searching is as follows:

Table 1: Hyperparamters for Random Search

Hyperparameter	Grid Search Space
Optimizers	['sgd', 'adam', 'adadelta']
Learning Rate	[i/10000 for i in range(0, 100)]
Dropout Rate	[0.3, 0.4, 0.5, 0.6, 0.7]
Gradient Clipping	[4.0, 4.5, 5.0, 5.5, 6.0]
Character Embedding Dimension	[20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]
Character LSTM Hidden State Dimension	[20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]

3.2 Transformer Model with PHI tags

Moving away from LSTM implement a Transformer-based model aiming to tackle this NER for patient notes task, with inspiration from a related adaptation called **TENER** [4].

3.2.1 Transformer Model

For clinical or patient notes, just like language, it is reasonable to believe that a deep bidirectional model is more powerful than either a left-to-right model or the shallow concatenation of a left-to-right and a right-to-left model. Therefore, we employ the same approach as the original BERT paper [6] and use A BERT-based model with associated weights. BERT is basically a trained Transformer Encoder stack. In an NER application, it would take a sentence in one language, and output related named-entities, or in our case, PHIs for each word in the sentence.

A Transformer model houses an encoding component, a decoding component, and connections between them. The encoding component is a stack of a pre-defined number of encoders. The decoding component is a stack of decoders of the same number. The encoder's inputs first flow through a self-attention layer – a layer that helps the encoder look at other words in the input sentence as it encodes a specific word. The outputs of the self-attention layer are fed to a feed-forward neural network. The exact same feed-forward network is independently applied to each position. The decoder has both those layers, but between them is an attention layer that helps the decoder focus on relevant parts of the input sentence. While we urge the reader to refer to the paper *Attention is All You Need* for finer details of the original proposal of The Transformer, some aspects of the model architecture particularly relevant to patient notes are:

- **Multi-headed attention:** A refinement of the self-attention layer that improves the attention layer and expands the model's ability to focus on different positions. This also gives the attention layer multiple "representation subspaces". With multi-headed attention we have not only one, but multiple sets of Query/Key/Value weight matrices. This notion is especially relevant for patient notes, given information better contextualized from each position's representation in such a dynamic form is likely to help discern PHIs from non-PHIs.

- **Relative Position Embedding:** An even more refined version of The Transformer adds a vector to each input embedding to meaningfully account for the order of the words in the input sequence. These vectors follow a specific pattern that the model learns, which helps it determine the position of each word, or the distance between different words in the sequence. The intuition here is that adding these values to the embeddings provides meaningful distances between the embeddings of various parts of clinical information once they're projected into Query/Key/Value vectors and during dot-product attention.

3.2.2 PHI Tags

In order to incorporate custom PHIs that are identified by the Transformer model, we implemented a model that takes as input a list a PHIs to function as tags for prediction. A tag of "0" is assigned to a token that identifies any given PHI. By describing ordered content of discharge note as sentences, PHIs as entities to be tagged and the entire discharge summary as a document, we facilitate the the use of multi-headed self-attention, positional encoding, and masked language models (MLM).

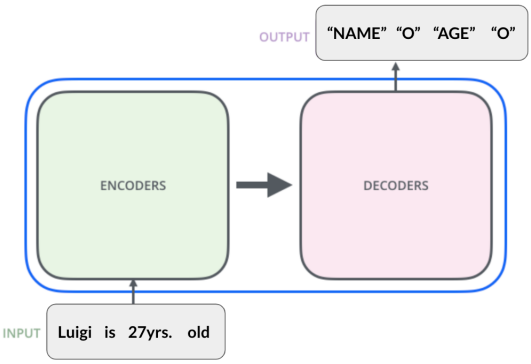

Figure 1: High-level input-output example for Anonymization task

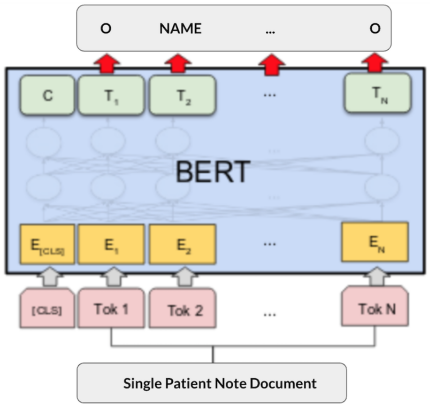

Figure 2: Model PHI-based output model, based on the original diagram from the BERT paper [6].

3.2.3 Simple Transformers

4 Experiments

4.1 Bi-directionsl LSTM & Random Search

4.1.1 Data

The CoNLL-2003 named entity data, which the given model and random search runs on, consists of eight files covering two languages: English and German [3]. For each of the languages there is a annotated data with a training file, a development file, a test file as well as a large file with unannotated data. The learning methods were trained with the training data. Our task focuses on the English language. Named entity recognition (NER) is a sub-task of information extraction (IE) that seeks out and categorises specified entities in a body or bodies of texts. Given this, any NER task requires annotated data to serve as labels that will be predicted. This is exactly what the aforementioned data provides, along with a file of uannotated text to further evaluate the model on.

4.1.2 Evaluation Method

In order to evaluate our models, we are using benchmarking metrics as provided by the NeuroNER paper. Our model is constructed to optimize accuracy over all other metrics, however, we use the other metrics to fine-tune the hyperparameters for the model. Let TP be the number of true positives, FP the number of false positives, and FN the number of false negatives. Then, the metrics are:

- Accuracy: Proportion of classifications that are correct, given by $\frac{TP+TN}{P+N}$
- Precision: Proportion of positive predictions that are gold labels, calculated given by $\frac{TP}{TP+FP}$
- Recall: Proportion of the gold labels that are correctly predicted, given by $\frac{TP}{TP+FN}$
- F1-score: With $b = \{0.5, 1, 2\}$, the harmonic mean of precision and recall, given by:

$$\text{F1} = \frac{(1+b^2)(\text{Precision} * \text{Recall})}{(b^2\text{Precision}) + \text{Recall}}$$

4.1.3 Experimental Details

- **Reproduction of NeuroNER results:** At the outset, we got a baseline model in the form of a TensorFlow implementation of **NeuroNER** developed by Dernoncourt et al. at MIT [2].
- **Random Search for Hyper-parameter Tuning:** For the baseline, we decided to use random hyperparameter search to fine-tune some of the default hyperparameters that are being used by the entity LSTM. Since we had the choice to change several different parameters, we decided to stick to changing six hyperparameters for ten iterations to observe significant differences in accuracy on the train, test and validation sets respectively.

Table 2: Summary of Results

optimizer	lr	dr	embed_dim	hidden_dim	epochs	grad_clip	Accuracy (Train)	Accuracy (Val)	Accuracy (Test)
sgd	0.0087	0.4	25	27	100	5.5	99.88%	98.71%	97.74%
adadelta	0.0006	0.3	21	21	125	4.5	83.24%	83.25%	82.41%
adadelta	0.0013	0.4	27	28	150	5.0	83.26%	83.25%	82.51%
adadelta	0.0079	0.5	24	23	125	5.5	89.00%	89.60%	89.40%
adam	0.006	0.3	30	24	125	4.0	96.70%	96.23%	94.93%
adam	0.0021	0.7	27	20	175	6.0	98.58%	97.96%	96.67%
adadelta	0.0026	0.4	28	25	125	4.0	83.31%	83.25%	82.54%
adam	0.0075	0.6	20	20	150	5.5	93.60%	93.48%	91.98%
sgd	0.0043	0.6	23	20	175	5.0	99.70%	98.72%	97.76%
sgd	**0.0021**	**0.7**	**23**	**29**	**100**	**5.0**	**99.53%**	**98.75%**	**97.86%**

4.1.4 Results

A brief summary of our results from running random grid search for ten iterations is displayed in table 2. The maximum number of epochs had no effect on the final accuracy, since the models converged way before they hit the maximum number of epochs every time. In general, models with higher learning rates yielded better results on average. Increasing the dropout rate also increased the accuracy on the three splits of the dataset. Increasing number of embedding dimensions increased the accuracy, however increasing embedding dimensions did not have any major effect on the results.

4.2 The Transformer

4.2.1 Data

The i2b2-NLP Challenge Dataset is made available by the Harvard Medical School [8]. It consists of annotated and un-annotated, de-identified patient discharge summaries. For each word in each summary, the word is assigned one of the 23 following categories, each of which represents a figment of personally identifiable information, or represents an ordinary word (`O`):

- `O`
- `PATIENT`
- `PROFESSION`
- `HOSPITAL`
- `DATE`
- `MEDICALRECORD`
- `CITY`
- `STATE`
- `ZIP`
- `COUNTRY`
- `FAX`
- `AGE`
- `IDNUM`
- `DOCTOR`
- `PHONE`
- `ORGANIZATION`
- `EMAIL`
- `STREET`
- `USERNAME`
- `DEVICE`
- `BIOID`
- `LOCATION-OTHER`
- `HEALTHPLAN`
- `URL`

The training, validation and testing split for our purposes is roughly a 50:15:35 split. Each document, as given in the dataset, is an `XML` file that presents the information given in a medical record along with words that wrapped in tags that denote a certain label as above. In order to feed the data as input to our transformer model, we converted the data found as `XML` files into a single training `TXT` file for each split of the dataset. This data preprocessing step has no effect on the final result, both qualitative and quantitative, and does not lead to an increase in the time-complexity of the model itself.

4.2.2 Evaluation Method

As described previously, we wanted to evaluate our models using benchmarking metrics as provided by the NeuroNER paper. Our model is constructed to optimize accuracy over all other metrics, however, we use the other metrics, such as precision and recall, to fine-tune the hyperparameters for the model. During the training procedure, we observed a steady decrease in the evaluation loss which was modeled as a standard cross-entropy loss function, which served as a sanity check to infer that the transformer model was training as expected.

4.2.3 Experimental Details

- **Training and Validating Transfomer with 1 epochs** We were interested in exploring the transformer model's performance when trained for increasingly longer time periods. Therefore, we began training the transformer for 5 epochs, and then evaluating on the validation set. This method also guided our search for hyperparamters that we should tune for better results on the training set.
- **Training and Validating Transfomer 5 epochs** We trained our model for 5 epochs and evaluated it on the validation set yet again, similar as before. Interestingly, we noticed a sharp dropoff of loss values until epoch 7, but then observed a very small decrease in the overall loss value, which was to be expected. For this run, we used `biobert` instead of `bert` as our pretraind language representation model.
- **Training and Validating Transfomer with 10 epochs** We trained our model for 40 epochs and evaluated it on the validation set yet again, similar as before.

- **Consolidating results** Eventually, we tested each of the models on the test set to consolidate final results and evaluate each model's performance and compare the benchmarks produced for quantitative and qualitative analysis of the results.

4.2.4 Results

A brief summary of our results from running the experiments listed above are displayed in Table 3. Our results show that the Transformer model performs better on the test set when it is trained for a smaller number of epochs (in terms of accuracy). The overall training time per epoch is reduced significantly when using a GPU, and even a simple transformer produces near-state-of-the-art accuracy results.

Table 3: Summary of Results

Set	Epochs	Eval Loss	Precision	Recall	F-1 Score	Accuracy
Training	1	0.020883835	0.9309826	0.9208633	0.9258953	0.9958265472
	5	0.010033232	0.9309826	0.8900042	0.9100323	**0.9963233432**
	10	0.009232332	0.9309826	0.8930022	0.9115969	0.9960002142
Validation	1	0.066309592	0.84833795	0.8128732	0.8302270	0.9891263775
	5	0.053810394	0.80389349	0.7833567	0.7934922	0.9834534354
	10	0.050019746	0.89340920	0.88324212	0.8882965	**0.9943242342**
Test	1	0.059244686	0.86247676	0.82806163	0.8449189	**0.9901509732**
	5	0.053434331	0.87110835	0.84288321	0.8567633	0.9832342332
	10	0.050232322	0.87221298	0.92130021	0.8960848	0.9862324521

5 Analysis

5.1 Quantitative Analysis

From a quantitative standpoint, we were concerned with accuracy scores over all three splits of the dataset for different epochs of training. We observed that the highest accuracy achieved on the test set was with just one epoch of training. This was a surprising result because more training usually implies better accuracy. However, we found that the necessary language substructures were identified by the Transformer model with a relatively short amount of training. Importantly, the model trained for 5 epochs performed best on the training set and the model trained for 10 epochs had the best performance on the validation set. Evaluation loss for all three models was constantly decreasing, with the sharpest decrease per iteration observed on the first model trained for just one epoch, which was to be expected.

5.2 Qualitative Analysis

We observed some interesting results on the medical NER task while using the Simple Transformer model. We believe that we were able to achieve near state-of-the-art results on the i2b2 challenge dataset even with relatively less amount of training because of the presence of a lot of words labelled as `O` in the dataset. Analyzing the output yielded by the model, it is clear that this disproportionate spread of labels in our dataset played a big factor in the large accuracy. However, the model accurately predicted many labels that were PHIs, such as `PATIENT` and **BIOID**. This was partly because of the presence of these PHIs concentrated in certain areas of the document structure, which may have been accurately detected by the Transformer model. Finally, we observed that the use of `biobert` over `bert` did not necessarily improve our model accuracy.

6 Conclusion

Recent developments in NLP techniques have indeed opened up a world of possibilities to be able to experiment with medical anonymization data. Towards that end, we utilize one such approach in the form of a Transformer model to carry out a PHI de-identification task.

Our work is limited in that we did not let any of our models train to full convergence,

which makes our results less comparable to those of state-of-the-art examples. Moreover, we were also constrained by time and computation resources that made it harder to evaluate our model on larger datasets, such as MIMIC-3 made available by MIT. However, for our baseline, we do hyper-parameter tuning for current state-of-the-art models over which we proceed with a Transformer approach. We achieve near state-of-the-art results with our Transformer model on the i2b2 dataset.

Future work should include additional hyper-parameter tuning on our current model and extended training time for each model, in order to make claims about these models compared to current state-of-the-art results. We would also like add other language representation models to see if there is an improvement in performance.

Nevertheless, our major contribution is showing the viability of a model that enables more parallelization and consequently scalability than existing state-of-the-art models towards such a task that is sure to shape the future of data sharing in the clinical domain.

References

[1] Franck Dernoncourt, Ji Young Lee, Peter Szolovits and Ozlem Uzuner. *De-identification of Patient Notes with Recurrent Neural Networks*. arXiv:1606.03475v1 [cs.CL] 10 Jun 2016.

[2] Dernoncourt, Franck and Lee, Ji Young and Szolovits, Peter. *An easy-to-use program for NER based on neural networks*. Conference on Empirical Methods on Natural Language Processing (EMNLP), 2017. `https://github.com/Franck-Dernoncourt/NeuroNER`.

[3] Erik F. Tjong Kim Sang and Fien De Meulder. *Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition*. Language Technology Group University of Antwerp, 2015.

[4] Hang Yan, Bocao Deng, Xiaonan Li and Xipeng Qiu. *TENER: Adapting Transformer Encoder for Named Entity Recognition*. `https://github.com/fastnlp/TENER`.

[5] MT-BioNER: Multi-task Learning for Biomedical Named Entity Recognition using Deep Bidirectional Transformers 01/24/2020 by Muhammad Raza Khan, et al. Amazon Microsoft share

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arxiv, 2018

[7] BEHRT: TRANSFORMER FOR ELECTRONIC HEALTH RECORDS

[8] i2b2: NLP Challenge Dataset Provided by the Harvard Medical School

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762 [cs], apr 2017.

[10] J Biomed Inform. 2017 Nov;75S:S34-S42. doi: 10.1016/j.jbi.2017.05.023. Epub 2017 Jun 1. De-identification of clinical notes via recurrent neural network and conditional random field. Liu Z1, Tang B2, Wang X3, Chen Q4.