User Response Classification Challenge

Adrien Cogny
Cornell Tech
ac2753@cornell.edu

April 21, 2018

Abstract

Abstract Text

I. Introduction

Chatbots are used in many applications today: customer support, flight booking, scheduling meeting, ordering food and many more. The application of a chatbot explored in this dataset is for a therapy chatbot. These types of chatbot, while very effective, may require human intervention. Determining when a human should intervene can be quite important, in this case when a person requires help in dealing with a complex situation, and requires tools to identify these situations.

The data set contains 80 examples of responses entered into a therapy chatbot. Each of these responses contains an id as well as an identification. The identification is either "flagged" if the response was flagged for human intervention or "not flagged" if not.

The task at hand was to create an AI agent to classify the user response.

II. Tools

The following tools and modules were used to complete this task:

- python 3.
- pandas
- dy-net (for the RNN)
- scikit learn (for the Random Forest)
- numpy
- tqdm (for progress bars)
- csv (For reading the csv embeddings to pandas)
- re (regular expressions) (for cleaning the data)

III. Preparing Data

The input data being sentences had to be cleaned up before passing into the models.

The first step was to load the csv file into a pandas dataframe and see what the data looked like. The data was, as mentioned above, a label as well as a sequence of words (not an array implementation yet). Due to the inherent nature of natural language processing both the label and sequence of words had to be converted to something which the machine could understand. That is, the label had to be converted from "flagged" or "not flagged" to 1 or 0 respectively and the sequence had to be converted to a series of word embeddings where each embedding represented a single word.

Natural language contains many words that are very common in sentences. These so-called stop words ("their", "he", "she", etc...) can make classifying a sentence very hard as, when combining word embeddings, they will take over the representation of the sentence simply by shear number. That is why in the preprocessing of each sentence (before it is given to the model), the stop words were removed from the sentence. By removing the stop words, we do not loose much important information and are able to classify more easily.

The models were created to do the conversion from label to 1 or 0 and from sentence of words to sequence of embeddings. The embeddings used were the GloVe 6B embeddings which come from wikipedia scapping[1].

IV. Models

i. RNN

When looking at sentence classification, one of the first thought was too look at an RNN encoder that would encode the sentence word by word and the computing a probability of being "flagged" or "not flagged". The label with the highest probability would then be applied to the sentence input.

i.1 RNN Description

DESCRIPTION OF RNN AND IMAGE/SKETCH OF THE MODEL CRATED

i.2 Tuning Parameters

When the model was created, the different parameters were tuned:

- Embedding Dimmension
- Hidden Dimmension Size
- Number of Epochs Run

The results for all of these tuning experiments are shown in the results section.

ii. Random Forest

After getting results for the RNN encoder and finding the best possible RNN, it was posited (based on research into text classification) that a random forest classifier could be more apt at this task.

[1] https://nlp.stanford.edu/projects/glove/

ii.1 Random Forest Description

RANDOM FOREST DESCRIPTION

Tuning Parameters When the model was created, the different parameters were tuned:

- Number of Estimators
- Sentence to Embedding Methodology

Whereas the number of estimators is a property of the random forest model itself, the sentence to embedding methodology describes how a sentences (or rather sequence of words) is transformed into a single vector wich can be input to the Random forest model.

There were two methods for embedding the sentence. One was to compute the mean of all the word embeddings and the other to compute the sum. The mean would attempt to construct a mean representation of the sentence using all the words in the sentence. The summation would create a sentence which was a sum of its parts.

V. Results

i. RNN

i.1 hidden dimmension

The hidden dimmension test was done by keeping all parameters of the RNN constant except for the hidden dimmension of the RNN. The following figure shows the results for the hidden dimmension test performed. The test was performed by changing the size of the embedding dimmension from 0 to 9 dimmensions in steps of 1. The training loss, dev set true positive , dev set true negative , dev set false positive and dev set false negative were computed for each of the models run and a graph was created showing the true positive rate and the false positive rate.

	type	test	filename	maxEpochs	num_layers	embeddingSize	hiddenDim	train_loss	accuracy	truePos	trueNeg	falsePos	falseNeg
0	RNN	hidden_dimm	hidden_dimm_stats0.csv	400	1	50	0	36.1499	0.716667	0	17.2	6.8	0
1	RNN	hidden_dimm	hidden_dimm_stats1.csv	400	1	50	1	5.5576	0.579167	2.2	11.7	5.5	4.6
2	RNN	hidden_dimm	hidden_dimm_stats2.csv	400	1	50	2	0.886681	0.504167	2.6	9.5	5.7	6.2
3	RNN	hidden_dimm	hidden_dimm_stats3.csv	400	1	50	3	0.00841726	0.591667	2.4	11.8	5.3	4.5
4	RNN	hidden_dimm	hidden_dimm_stats4.csv	400	1	50	4	0.00756414	0.483333	2	9.6	5.1	7.3
5	RNN	hidden_dimm	hidden_dimm_stats5.csv	400	1	50	5	0.00441678	0.529167	1.9	10.8	4.9	6.4
6	RNN	hidden_dimm	hidden_dimm_stats6.csv	400	1	50	6	0.00633066	0.5375	1.6	11.3	4.6	6.5
7	RNN	hidden_dimm	hidden_dimm_stats7.csv	400	1	50	7	0.00421989	0.558333	1.9	11.5	5.8	4.8
8	RNN	hidden_dimm	hidden_dimm_stats8.csv	400	1	50	8	0.00384365	0.5125	1.9	10.4	6.1	5.6
9	RNN	hidden_dimm	hidden_dimm_stats9.csv	400	1	50	9	0.0032208	0.491667	2.3	9.5	5.2	7

Figure 1: *Shows the table of hidden dimmension tests for the RNN*

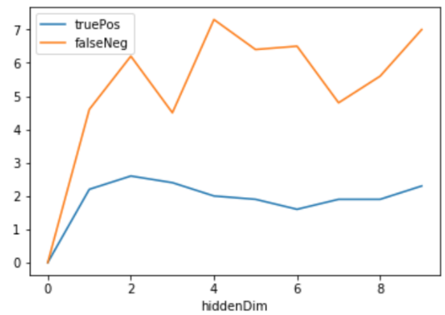

Figure 2: *Shows the graph of true positive count vs hidden dimmension of RNN*

This graph shows that the true positive count is always lower than the false negative count. This means that there are more instances where the model will classify a sentence as "not flagged" when in fact it should be "flagged" than there are instances where the model correctly classifies a sentence as "flagged." While this points towards this particular model (with the hyperparameters described below) not being good, the true positive rate and accuracies are derived (and much more important) metrics to look at.

While this graphs shows the true positive count as well as the false negative counts, a more interesting metric which can be derived fromt he true positive count and the false negative count is the true positive rate (tpr) which shows how much of the truth the model captures.

The following figure shows the tpr for the RNN model for different hidden dimmensions.

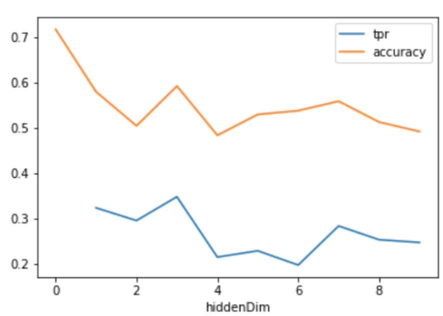

Figure 3: *Shows the graph of true positive rate vs hidden dimmension of RNN*

From this figure, we can clearly see that the tpr is greatest when the hidden dimmension is 3, with the other parameters set to: number of epochs ran = 400, number of layers = 1 and embedding size = 50. This model received an accuracy of 0.591667 and a true positive rate of 0.347826.

While the graph of accuracies shows that when the hidden dimension is of size 0, the accuracy jumps to 0.7, this model would not be considered to be a good model as the true positive rate is non existant because we are not classifying any results as being "flagged", which defeats the whole purpose of the model.

Thus, the hidden dimmension will be set to 3 for the other models.

i.2 GloVe embedding size

When the hidden dimmension was found, the next hyper-parameter which could be tuned was the size of the GloVe embeddings used to conver the sentences into something the models could understand.

This next figure shows the true positive counts and false negative counts for each embedding dimmension available (50, 100, 200 and 300).

type	test	filename	maxEpochs	num_layers	embeddingSize	hiddenDim	train_loss	accuracy	truePos	trueNeg	falsePos	falseNeg
RNN	embedding_dimm	embedd_size_stats50.csv	400	1	50	3	0.0081134	0.566667	2	11.6	5.26667	5.13333
RNN	embedding_dimm	embedd_size_stats100.csv	400	1	100	3	0.00727635	0.494444	2.26667	9.6	5.26667	6.86667
RNN	embedding_dimm	embedd_size_stats200.csv	400	1	200	3	0.00961309	0.591667	2.86667	11.3333	5.06667	4.73333
RNN	embedding_dimm	embedd_size_stats300.csv	400	1	300	3	0.00750879	0.505556	2.4	9.73333	5.4	6.46667

Figure 4: *Shows the table of embedding dimmension tests for the RNN*

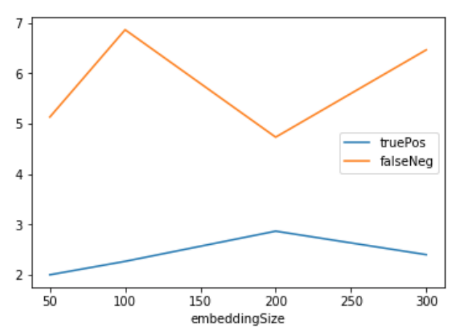

Figure 5: *Shows the graph of true positive count and false negative count vs embedding dimmensions of RNN*

The table and graph show that the true positive count is always smaller than the false negative count. However a dip can be seen in the false negative count

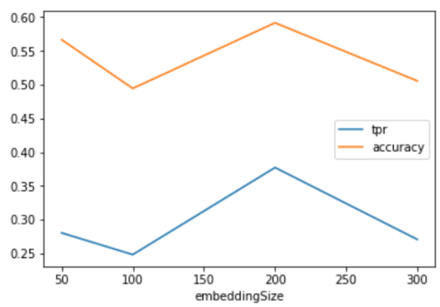

Figure 6: *Shows the graph of true positive rate vs embedding dimmensions of RNN*

i.3 Number Layers

ii. Random Forest

Table 1: *Example table*

Name		
First name	Last Name	Grade
John	Doe	7.5
Richard	Miles	2

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

$$e = mc^2 \tag{1}$$

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

VI. Discussion

i. Subsection One

A statement requiring citation [?]. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

ii. Subsection Two

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

References

[Figueredo and Wolf, 2009] Figueredo, A. J. and Wolf, P. S. A. (2009). Assortative pairing and life history strategy - a cross-cultural study. *Human Nature*, 20:317–330.