CSCE 638 Natural Language Processing
A Project Report on

Sequence Classification using Semi-Supervised Learning

Ankur Kunder (UIN: 729007628), Gehao Yu (UIN: 629008717)
Department of Computer Science,
Texas A&M University,
College Station, Texas, USA

Introduction

Sequences are a collection of elements or objects such that each element is related to its position in the sequence, and also to the other elements in the sequence. Classification is a task that assigns labels to the input from the given output classes. Therefore, Sequence classification is a task that deals with the problem of assigning a label to a sequence of inputs over space and time. Sequence classification is a challenging problem because it requires the classifier to consider sequences of variable length, while also considering the long term relations in between the elements. In addition to this, the input elements could belong to a very large vocabulary.

In classification using Machine Learning models with Supervised Learning, there is always a strong dependence on the availability of a large amount of labeled data. For text classification, labeled data can be particularly hard to find, and in addition to this, it's costly to train human annotators for labeling the samples. But, if we have a few example samples with labels and a large amount of unlabeled data, we can use semi-supervised learning methods to achieve significant accuracy on classification tasks.

Through our project, we attempt to learn about the classification labels for paragraphs, i.e. text data. In doing so, we train two classifiers for binary classification and multi-class classification of paragraphs. We test the limits on the amount of labeled data needed for best performance on our datasets. Apart from this, we also focus on LSTMs and their use for text classification. In doing so, we try two different Models, a vanilla LSTM and a Bi-directional LSTM. We also report the cases that were difficult to classify and briefly discuss the reasons behind them.

Related work

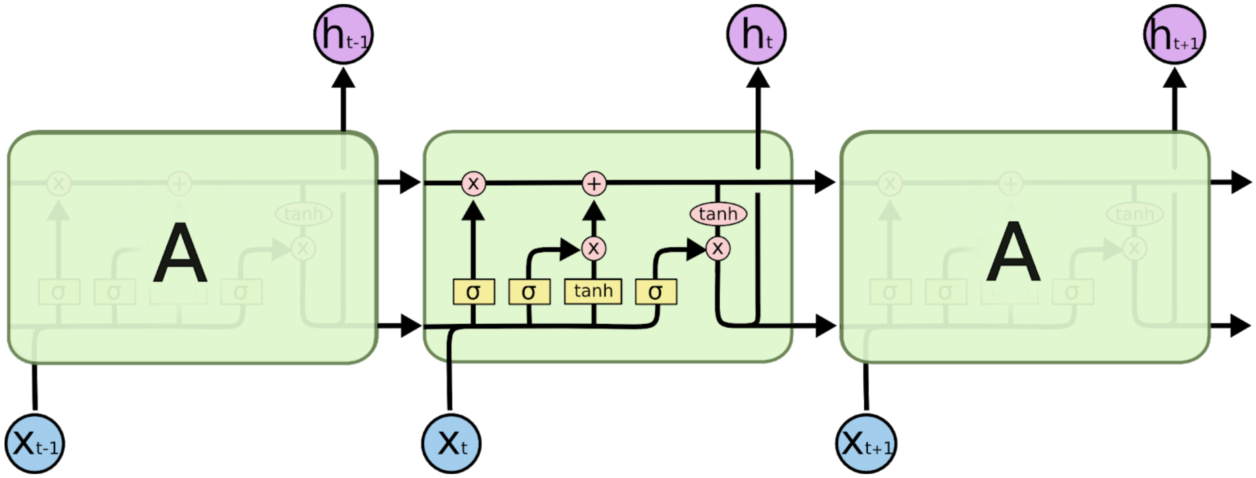
BOG(bag of words) is a basic document representation model method. BOG shall transform, for example, a sentence, into a dictionary. If given a sentence like ‘John likes to watch movies. Mary likes too.’ and we create a dictionary {"John": 1, "likes": 2, "to": 3, "watch": 4, "movies": 5, "also": 6, "football": 7, "games": 8, "Mary": 9, "too": 10}. Then the sentence turned into a vector [1, 2, 1, 1, 1, 0, 0, 0, 1, 1]. But this method also has obvious disadvantages: we completely lose all the sequential information in the sentence, which are important especially in nature language model.

SVM can be used in text classification problems as well. The process can be concluded as:

1) Segment the words in the documents, and use the words as the vectors
2) After transforming into vectors, we have to find out the most important features. There are many methods, including counting vectors, TF-IDF vectors, word embedding vectors, etc.
3) Based on the features we selected, we shall train a classifier using SVM to find out the best hyperplane and test on the dataset.

There are also some inadequacies with SVM, one of them is that SVM can hardly deal with the multi-class classification problems. Usually people combined several binary classifiers to deal with multi-class classification problems, but its performance is not always satisfying.

RNN were an obvious choice for text classification prior to the use of LSTMs, because they are able to learn from sequential data due to the feedback loop of hidden state. But, they had the problems of gradient decay and gradient explosion, because of gradient multiplication for each backpropagation step. As a result of this, RNNs tend to suffer from Short-term memory, i.e. they are unable to learn long-term context and relations between elements/words.
These problems are alleviated by the use of an LSTM, with ReLU activation and gradient clipping.



LSTMs (shown above) are a type of Recurrent Neural Network that have a cell state in addition to the hidden state. In each subsequent step, the cell state value, and the hidden state values are controlled by the use of three gates - a forget gate, an input gate, and an output gate. The equations for these gates are as below:

$$f_t \;=\; \sigma(W_f * \;[h_{t-1}, x_t] \;+ b_f)$$
$$i_t \;=\; \sigma(W_i * \;[h_{t-1}, x_t] \;+ b_i)$$
$$C'_t \;=\; tanh(W_c * \;[h_{t-1}, x_t] \;+ b_c)$$
$$C_t \;= f_t * C_{t-1} \;+ i_t * \; C'_t)$$
$$o_t \;=\; \sigma(W_o * \;[h_{t-1}, x_t] \;+ b_o)$$
$$h_t \;= o_t * tanh(C_t)$$

Here, *x* is the input, *f* is the forget gate, *i* is the input gate, *o* is the output gate, *C* is the cell state and *h* is the hidden state. An LSTM is able to learn long term context from the sequences due to better gradient flow from one cell to previous cell [6].

In Semi-Supervised Learning, a small amount of labeled data is used in conjunction with large unlabeled data to get improvements in the learning accuracy. The costs associated with acquiring labeled data by training a human annotator or conducting an experiment, may render a fully labeled dataset infeasible. But, unlabeled data is inexpensive and often available in abundance, and therefore, Semi-Supervised Learning in such cases can have greater practical value. Transductive and Inductive learning are two broad classifications of Semi-Supervised Learning[1]. In Transductive learning, labels for unlabeled data are learnt. Whereas, in Inductive learning the mapping from labeled data to it’s labels is learned.

Datasets Used

1) IMDB 50K Movie Review Dataset

The IMDB Movie Review Dataset[2] is for Binary Sentiment Classification. It has 50K movie reviews with equal number of positive and negative reviews. A sample example is given below:

I thought this was a wonderful way to spend time on a too hot summer weekend, sitting in the air conditioned theater and watching a light-hearted comedy. The plot is simplistic, but the dialogue is wi...	positive
Basically there's a family where a little boy (Jake) thinks there's a zombie in his closet & his parents are fighting all the time.<br /><br />This movie is slower than a soap opera... and suddenly, J...	negative

2) 20 Newsgroups

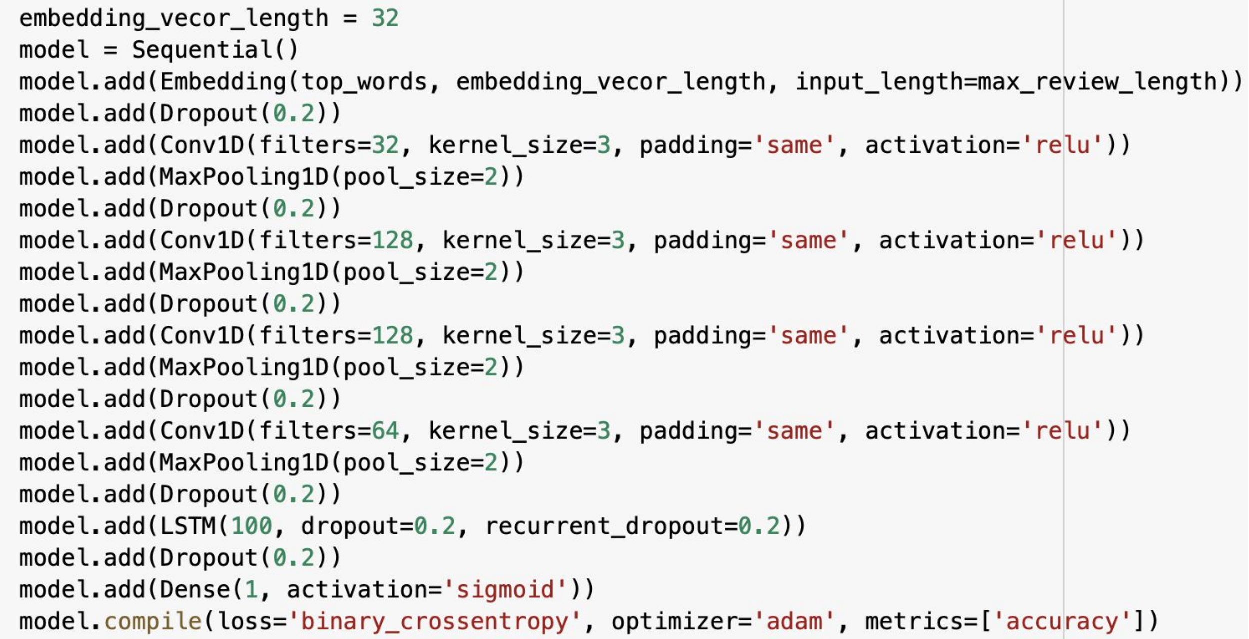
This data set is a collection of 20,000 messages, collected from 20 different netnews newsgroups. One thousand messages from each of the twenty newsgroups were chosen at random and partitioned by newsgroup name. The list of newsgroups from which the messages were chosen is as follows :

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.crypt sci.electronics sci.med sci.space
misc.forsale	talk.politics.misc talk.politics.guns talk.politics.mideast	talk.religion.misc alt.atheism soc.religion.christian

Our Methods

1) LSTM for Binary Classification with Semi-Supervised Learning(Self Training)

The model used for Binary Sentiment Classification of Movie Reviews consists of an Embedding Layer, 4 CNN layers, and an LSTM which was then followed by a fully connected layer to the output. The same has been described below in detail:



```
embedding_vecor_length = 32
model = Sequential()
model.add(Embedding(top_words, embedding_vecor_length, input_length=max_review_length))
model.add(Dropout(0.2))
model.add(Conv1D(filters=32, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(Conv1D(filters=128, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(Conv1D(filters=128, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(Conv1D(filters=64, kernel_size=3, padding='same', activation='relu'))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

The input for the model is preprocessed by converting the sequence text into lowercase words, getting the word tokens, and then replacing the words by their frequency. Only the top 5000 most frequent words are considered and the maximum length of the sequence is fixed at 500. Therefore, the infrequent words are replaced by <UNK> token and the sequences are either padded with <UNK> or clipped to satisfy the maximum length. The Embedding Layer tries to learn a 32 length vector for each word and then Conv1d is applied over it with filter size of 3x3. This is followed by the LSTM Layer which outputs a 100 length vector, after processing a sequence. Finally, a fully connected layer with Sigmoid activation is used to output the label between 0 and 1. The model learns to decrease the binary cross-entropy loss using the Adam update for updating weights.

The above model was made after experimenting with different number of CNN layers, sizes of filters, the use of dropout, and the number of fully connected layers. The accuracy improved with increase in the number of CNN layers till 4 layers and dropped after that. The accuracy improved with the inclusion of dropout layers, as it makes the model more sparse. The addition of fully connected layers after the LSTM decreased the accuracy of the model.

For Semi-Supervised Learning by Self-Training, the unlabeled data is classified by the classifier that has learned from the labeled data for two epochs. Those samples which are confidently classified by the classifier are included in the next round of training as labeled data, with the predicted class as labels. This process is repeated until acceptable accuracy is reached or until the entire unlabeled data has been confidently classified.

The confidence is defined by the threshold value for prediction, which is taken as 0.2. If a prediction is less than 0.2, it is classified as 0, and if it is greater than 0.8, it is classified as 1, else the unlabeled data is not included in the next round of training.

2) Bidirectional LSTM for Multi-class Classification with Semi-Supervised Learning

Bidirectional-LSTM is the method that I used in 20 newsgroup dataset. Different from the simple LSTM block which only considering former words, Bidirectional-LSTM reverses the sentence sequence and gained the information of latter words as well.

For example, if we send the sentence vector ’I love burrito.’ into Bi-LSTM block, we will get three vectors $\{h_{L0}, h_{L1}, h_{L2}\}$ with forward $LSTM_L$ ‘I’ ‘love’ ‘burrito’. Also, the backward $LSTM_R$ entered ‘burrito’ ‘love’ ‘I’ received three vectors $\{h_{R0}, h_{R1}, h_{R2}\}$. We can concatenate two vectors into one as $\{[h_{L0}, h_{R2}],[h_{L1}, h_{L1}],[h_{L2}, h_{L1}]\}$ called$\{h_0, h_1, h_2\}$, but this combined vector is too complicated, in my model I only take $[h_{L1}, h_{R2}]$.

When it comes to semi-supervised learning part, same as Ankur, I also used self training algorithm. Firstly, I separated my dataset into two parts: one is training data and one is testing data. The early stopping threshold I set is 4 iterations, which means if test accuracy didn’t improve for 4 iterations, the training process shall stop.

There is 9606 labeled data in my training dataset, and I started with fully supervised training dataset. I arrived at about 90% test dataset accuracy with fully supervised method and about 80% with semi-supervised method, then each time I shall remove some data(1065 data each time) labels from training dataset.

For supervised learning, it means that the training dataset reduced 1065 samples each time, and for semi-supervised learning, it means that training dataset reduced 1065 labelled samples and added 1065 un-labelled samples each time.

The final result and analysis shall posted as following.

Results and Analysis

1) Analysis for LSTM model for Binary Classification with Semi-Supervised Learning(Self Training)

To draw conclusions about the use of Semi-Supervised learning method, the data was divided into three parts: labeled data, unlabeled data, and the test data. Supervised learning, when the 80% labeled data is entirely used for training is the upper bound of the accuracy that can be achieved from this model. Whereas, using only a small percentage (10%) of labeled data for Supervised learning and no use of unlabeled data will be our base mark for the accuracy using this model. Now, as we add more unlabeled data in the Self Learning step, the accuracy should improve. The results from the model are as below:

Learning Method	Labeled Data	Unlabeled Data	Test Data	Accuracy
Supervised Learning	80%	0%	20%	90.25%
Supervised Learning	10%	0%	20%	70.20%
Semi-Supervised Learning	10%	70%	20%	**85.38%**

By using additional unlabeled data, the accuracy on the test set is significantly improved. Therefore, the intended goal of Semi-Supervised Learning is accomplished.

The cases that the model incorrectly classifies, either had sarcastic comments, or were positive reviews about negative films, or lacked sufficient context. For example -

```
<START> <UNK> the hospital comedy <UNK> a series of <UNK> has <UNK>
manhattan hospital patients are dying left and right due to <UNK>
<UNK> and a <UNK> staff when a resident doctor is caught up in the
death count the chief medical <UNK> dr george c scott is called in
to investigate having worked as a doctor for too many years and
going through a mid life crisis of his own dr finds the going tough
he decides to commit suicide but then he meets barbara diana <UNK> a
young hippie beauty whose <UNK> <UNK> on life help the depressed br
br <UNK> black comedy features a de <UNK> performance from veteran
actor george c scott he's good at playing high <UNK> serious
characters whose <UNK> <UNK> are <UNK> <UNK> first half of the film
unfolds like a melodrama giving a pretty good account of hospital
life and the <UNK> they sometimes are but then as things look set
for a dramatic climax it <UNK> into slapstick comedy if <UNK> script
had <UNK> its dramatic feel i wonder if scott would've walked out
with another best actor oscar he had previously won it <UNK> the
year before his <UNK> suicide scene is one of the most <UNK> <UNK>
real in cinema history br br quote dr last night i sat in my hotel
room <UNK> the <UNK> of my life and <UNK> suicide i said <UNK> don't
do it you're a doctor a <UNK> you're a necessary person you're life
is then i find out that one of my <UNK> was killed by a couple of
<UNK> how am i to <UNK> my feeling of <UNK> in the face of thisTrue
```

Label: 1, **Predicted Label:** 0 **Reason:** A number of negative words have been used. Maybe with a longer length of sequence this will be correctly classified. A clear stance is not taken in the current context.

The model is used to classify a review from outside the dataset.

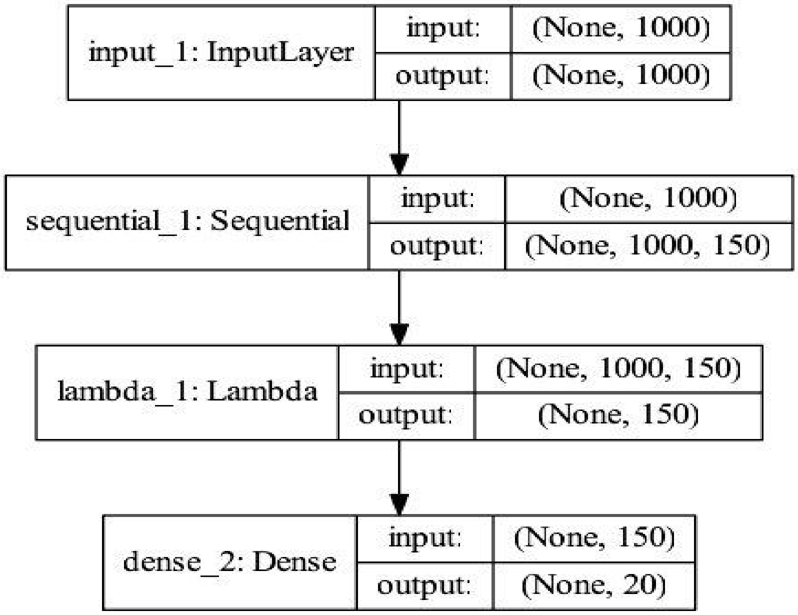
Review: “the movie was so bad i walked out of the theatre after an hour . the acting and dialogs were abysmal . there was no story to be followed . this was a waste of time .”

Processed input: <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <PAD> <START> the movie was so bad i walked out of the theatre after an hour <UNK> the acting and dialogs were abysmal <UNK> there was no story to be followed <UNK> this was a waste of time <UNK>

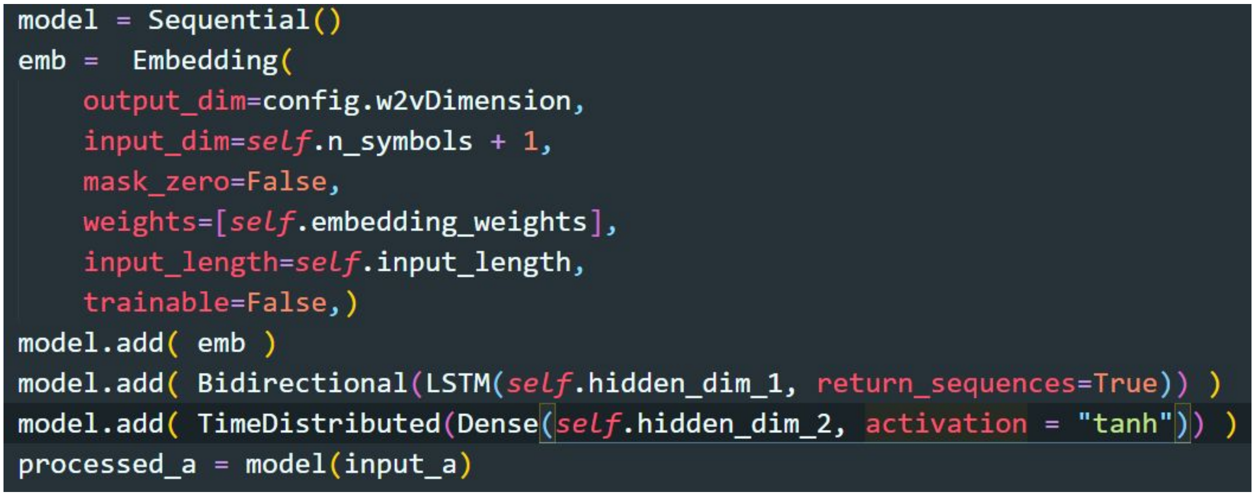
Predicted Label: [[1.1290014e-05]]

2) Analysis for Bidirectional LSTM model for Multi-class Classification with Semi-Supervised Learning

The dataset in total number is 20% test and 80% train, and among the train dataset, the least semi-supervised dataset ration is 15%, which means at least I only use 80%*15% data to train the model with semi-supervised method. All these information can be archived in config.py
The maximum training epochs I set is 100, but early stopping iteration I set is 4, so generally 20-40 epoches(It depends on how many training data I used) is enough.
The model structure I I trained is like this:

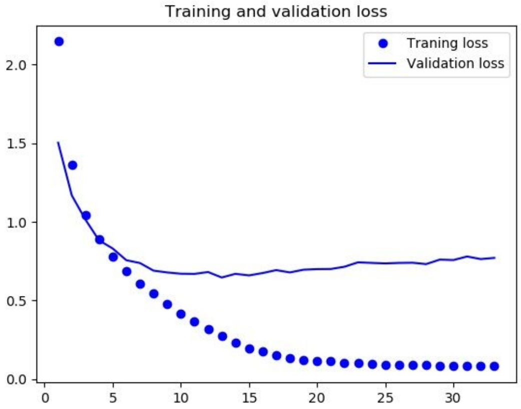


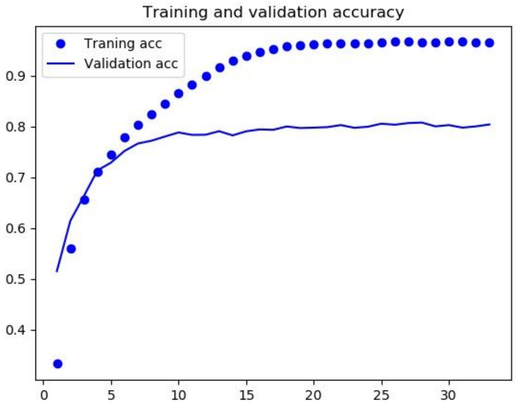
And the Sequential layer structure is one 150 units Bidirectional-LSTM layer and one 150 units fully connected layer:



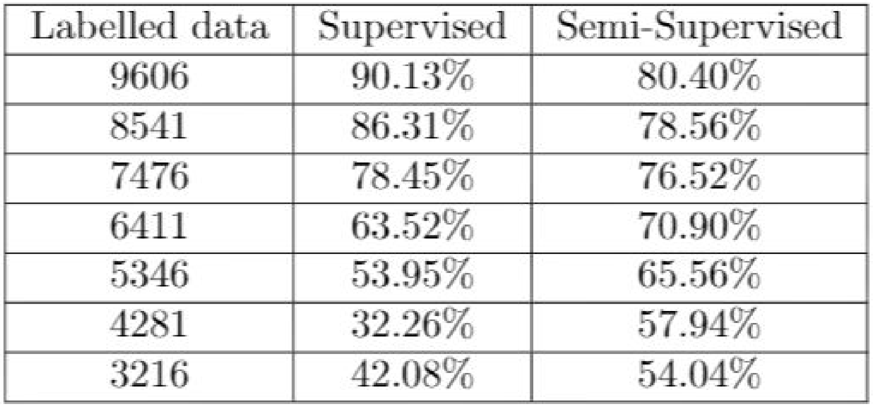
```
model = Sequential()
emb =  Embedding(
    output_dim=config.w2vDimension,
    input_dim=self.n_symbols + 1,
    mask_zero=False,
    weights=[self.embedding_weights],
    input_length=self.input_length,
    trainable=False,)
model.add( emb )
model.add( Bidirectional(LSTM(self.hidden_dim_1, return_sequences=True)) )
model.add( TimeDistributed(Dense(self.hidden_dim_2, activation = "tanh")) )
processed_a = model(input_a)
```

During the fully supervised learning process, the loss function and the accuracy performance trend is like

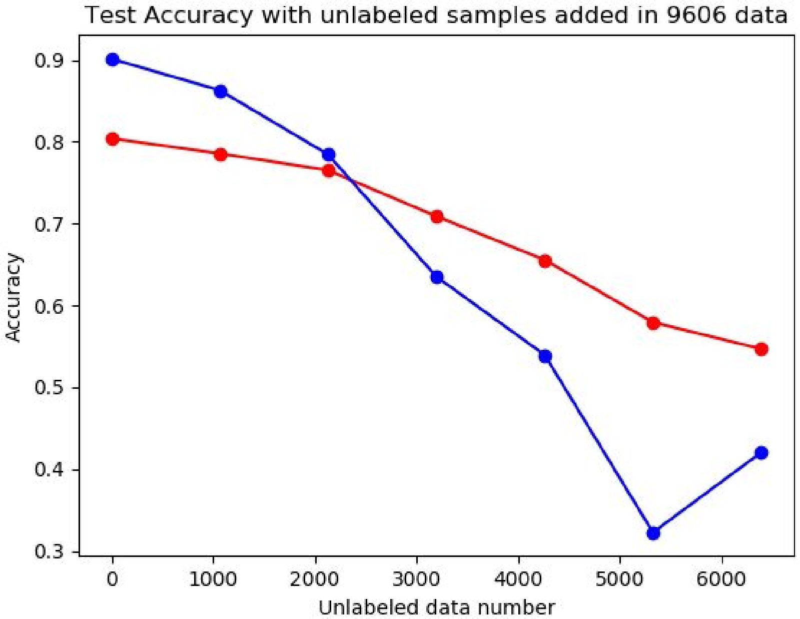




After the fully supervised learning process, I re-trained the model with different number of labelled data and unlabelled data. For supervised learning, the number of labelled data varied 9606 to 3216; for semi-supervised learning, the number of labelled data also varied from 9606 to 3216, but the number of total training data remained 9606, which means unlabelled data improved from 0 to 6390. And their test dataset performances posted as following:



Labelled data	Supervised	Semi-Supervised
9606	90.13%	80.40%
8541	86.31%	78.56%
7476	78.45%	76.52%
6411	63.52%	70.90%
5346	53.95%	65.56%
4281	32.26%	57.94%
3216	42.08%	54.04%



Also, I posted the top 10 key-words in different newsgroup based on TF-IDF index, for different topics, their most important words are posted as following:

```
alt.atheism: moral believ peopl think religion atheist wa say god thi
comp.graphics: thank know format use program ani thi file imag graphic
comp.os.ms-windows.misc: program win run thi os driver font file use window
comp.sys.ibm.pc.hardware: board mb control pc ide use scsi thi card drive
comp.sys.mac.hardware: ani mhz quadra use monitor drive problem thi appl mac
comp.windows.x: ani file motif program xterm widget display thi use window
misc.forsale: drive condit pleas includ price new sell offer ship sale
rec.autos: new like auto drive dealer ani wa thi engin car
rec.motorcycles: helmet like drive dog dod thi wa motorcycl ride bike
rec.sport.baseball: pitcher hit win hi wa player pitch year team game
rec.sport.hockey: playoff year thi espn player play wa hockey team game
sci.crypt: phone nsa use govern secur chip clipper thi encrypt key
sci.electronics: ha electron copi work anyon circuit power know thi use
sci.med: edu know msg ani ha patient doctor wa diseas thi
sci.space: use moon think satellit launch nasa wa orbit thi space
soc.religion.christian: christ say hi sin jesu church wa thi christian
god
talk.politics.guns: law govern right firearm weapon peopl fbi thi wa gun
talk.politics.mideast: kill say peopl thi jew wa isra arab armenian
israel
talk.politics.misc: ha make law govern men drug homosexu peopl wa thi
talk.religion.misc: kent object moral peopl koresh christian thi say wa
      god
```

Conclusion

As seen from our results, the inclusion of unlabeled data in training by the use of Semi-Supervised Learning, for cases where labeled data is insufficiently available, will result in an improvement in Test Accuracy. The models trained were able to classify the Test Sequences with acceptable accuracy and performed well on unseen samples.

References

1. Semi-Supervised Learning Literature Survey (2006)
2. IMDB 50K Movie Review Dataset
3. https://www.aclweb.org/anthology/W99-0908/
4. https://www.cs.bgu.ac.il/~elhadad/nlp17/Classification_20Groups_Sklearn.html
5. https://arxiv.org/abs/1605.07725
6. Understanding LSTM Networks

Work Done by Individual Team Member:

Ankur Kunder: Implemented the LSTM model for Binary Classification with Semi-Supervised Learning(Self Training). For the report, wrote the Introduction, Related work for RNN, LSTMs, Semi-Supervised Learning, the parts related to the method implemented, and the Conclusion.
Gehao Yu: Implemented the Bidirectional LSTM model for Multi-class Classification with Semi-Supervised Learning. For the report, wrote the Related work for Bag-of-Words, SVMs, and the parts related to the method implemented.