Sentiment Analysis of Filipino Tweets Using Recurrent Neural Network

Rusty John Lloyd Mina
University of the Philippines
Electrical and Electronics Engineering Institute
rusty.mina@eee.upd.edu.ph

Rowel Atienza
University of the Philippines
Electrical and Electronics Engineering Institute
rowel@eee.upd.edu.ph

ABSTRACT

Sentiment analysis is the process of determining the emotion expressed by a text, usually classified to either negative, neutral or positive. In this paper, the researchers propose the use of Recurrent Neural Network (RNN), specifically the Long Short-Term Memory Neural Network (LSTM), in analyzing sentiments of Filipino tweets. The network is modeled to have only binary classification, either the text contains negative or positive sentiment. The network was trained with 4108 tweets, 42% of which are positive while 58% are negative. It was then evaluated with 381 tweets. The network yielded a training accuracy of 97.29% and testing accuracy of 73.48%. The model can be implemented in just 5 lines of Python code in Keras with Tensorflow[1] as backend.

KEYWORDS

Sentiment Analysis, LSTM, RNN, NLP, Deep Learning

1 INTRODUCTION

Humans communicate through different means. We learn new information by reading news and texts describing the events happening around us. One of the information we get from reading is the sentiment being expressed by the text. Sentiment analysis is a well-known problem in Natural Language Processing (NLP) that seeks to determine the writer's attitude from a piece of text. If done properly, sentiment analysis can be an excellent source of information on marketing strategies, customer service and many more [3]. For instance, by knowing the public sentiment on a product, a company can know the strengths and weaknesses of their product and therefore improve it. An example of which is where Pagolu et al. [6] used sentiment analysis to correlate twitter responses on companies,with the movements of prices in stock market.

Humans can easily understand the meaning of a text however it may not always be the case for machines. Human language is elaborate, with nearly infinite grammatical variations, misspellings, slangs and other challenges making accurate automated analysis of natural language quite difficult [10]. So far, existing approaches that deal with the sentiment analysis of Filipino language either have low prediction accuracy or that the data used for training are imbalanced. In this paper, we introduce a deep-learning approach to sentiment analysis that is the same time flexible and relatively more accurate, i.e. it handles a large variety of Filipino tweets while accurately predicting the sentiment the text expresses. Our approach builds upon the IMDB sentiment analysis LSTM approach of Keras [2].

1.1. RELATED WORKS AND CONTRIBUTIONS

The current state-of-the-art sentiment analysis algorithms for Filipino language use machine learning technique called Naïve Bayes Classifier. Naïve Bayes is very simple and straightforward to use. Naïve Bayes algorithm assumes that features are independent of each other even though they are not, hence the name Naïve. Classification is then done based on the frequency of features and their prior probabilities. In the case of texts, the words are the features. For instance, we want to find the probability that "hello world foo" expresses a positive sentiment. We would find probability of the word "hello" appearing in positive texts. We do the same for the words "world" and "foo". We would multiply all three probabilities and then further multiplied by the probability of getting positive sentiments. Same is done in getting the probability of the text expressing negative sentiment. Whichever label has the higher posterior probability becomes the classification of the text. This technique was used by Pippin et al. [4] in classifying emotions expressed by Filipino tweets. However, the dataset used to train their classifier contains very skewed frequency of data. Rennie et al. [5] showed that skewed dataset causes the Naïve Bayes classifier to unwittingly prefer one class over the other and therefore.

Naïve Bayes was also used by Patacsil et al. [7] in sentiment analysis of customer reviews of Filipino ISPs. However, instead of directly applying Naïve Bayes algorithm, they first translated the texts to English using Google Translate API. The translation

may have introduced loss in overall meaning of the texts.

The major limitation of Naïve Bayes in sentiment analysis is its assumption that the words are strongly independent of each other. In the perspective of Naïve Bayes, the phrases "person of the year" and "year of the person" are most likely equal. Deep learning offers two architectures in dealing with this problem, the Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN). There are several researches on sentiment analysis of different languages that used CNN and RNN. Deriu et al. [8] proposed that use of CNN in sentiment analysis of tweets. They showed that CNN can achieve higher scores than shallower architectures like the traditional Naïve Bayes Classifier and State Vector Machine (SVM). Liu et al. [9], proposed the use of RNN for text classification. They showed that their RNN models outperform common models like SVM in text classification.

In this paper, we use Recurrent Neural Network specifically the Long Short-Term Memory in dealing the limitations of Naïve Bayes Classifier. From a practical perspective, our approach is an effective algorithm for sentiment analysis. RNNs are family of neural networks that are specialized for processing sequences [11]. Sentences (and tweets) are sequences of words therefore using RNN architecture is a natural choice.

2 RECURRENT NEURAL NETWORKS

In this section, we give an overview of the simple Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network that were used in the research.

Recurrent Neural Network (RNN). RNNs are called *recurrent* because they perform the same task for every element of a sequence, with the output being depended on the previous computations [12]. Another way to think about RNNs is that they have a "memory" which captures information about what has been calculated so far [12]. As said before, sentences can be treated as sequences of words.

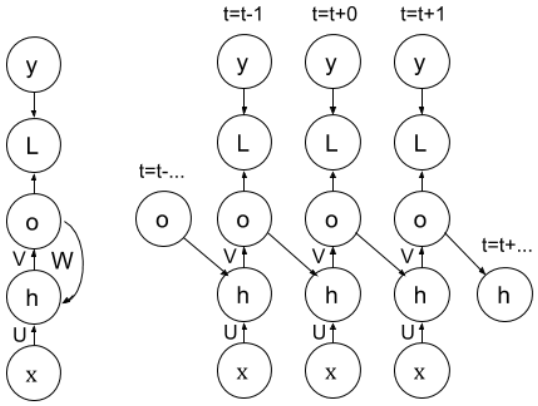

Fig. 1 A Simple RNN Cell and its Unfolding, adapted from [11]

As seen from Figure 1, the current step depends on the output of the previous step which also has input that depends on the output of the step before it, therefore making the current step as function of previous steps. The mathematical processes behind a RNN cell are:

$$\boldsymbol{a}^t = \boldsymbol{b} + \boldsymbol{W}\boldsymbol{o}^{t-1} + \boldsymbol{U}$$

$$\boldsymbol{h}^t = tanh(\boldsymbol{a}^t)$$

$$\boldsymbol{o}^t = \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^t$$

$$\boldsymbol{y}^t = sigmoid(\boldsymbol{o}^t)$$

where at each time step t, x^t is the input, h^t is the hidden layer activation, o^t is the output, y^t is the target output and L^t is the loss between y^t and o^t. Since we're only doing binary classification, sigmoid is used as activation for output. Such variant of RNN is less powerful and can only express a small set of functions. [11].

Long Short-Term Memory (LSTM). LSTM is a special type of RNN that is designed to avoid long-term dependency problems which the simple RNN has problem with [15]. LSTM can control, add or drop information of a cell state as regulated by its structures called gates.

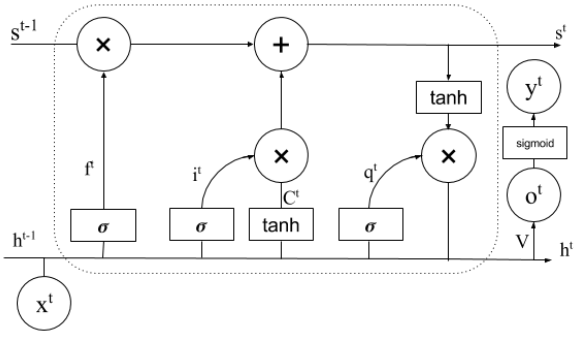

Fig. 2. A LSTM Cell, adapted from [14]

$$\boldsymbol{i}^t = \boldsymbol{\sigma}(\boldsymbol{W}_i \boldsymbol{x}^t + \boldsymbol{U}_i \boldsymbol{h}^{t-1} + \boldsymbol{b}_i)$$

$$\boldsymbol{C}^t = \tanh(\boldsymbol{W}_c \boldsymbol{x}^t + \boldsymbol{U}_c \boldsymbol{h}^{t-1} + \boldsymbol{b}_c)$$

$$\boldsymbol{f}^t = \boldsymbol{\sigma}\left(\boldsymbol{W}_f \boldsymbol{x}^t + \boldsymbol{U}_f \boldsymbol{h}^{t-1} + \boldsymbol{b}_f\right)$$

$$\boldsymbol{s}^t = \boldsymbol{i}^t \boldsymbol{C}^t + \boldsymbol{f}^t \boldsymbol{s}^{t-1}$$

$$\boldsymbol{q}^t = \boldsymbol{\sigma}\left(\boldsymbol{W}_q \boldsymbol{x}^t + \boldsymbol{U}_q \boldsymbol{h}^{t-1} + \boldsymbol{b}_q\right)$$

$$\boldsymbol{h}^t = \boldsymbol{q}^t \tanh(\boldsymbol{s}^t)$$

Shown above are the mathematical processes done in the gates between $\boldsymbol{h}^t$ and $\boldsymbol{x}^t$ of an LSTM cell. $\boldsymbol{i}^t$ models the input gate, $\boldsymbol{c}^t$ is the candidate value for the states of the current memory cell, $\boldsymbol{f}^t$ is the activation of forget gate, $\boldsymbol{s}^t$ is the new value of the state and $\boldsymbol{q}^t$ is the value of the output gate. To get the memory cell output:

$$\boldsymbol{o}^t = \boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}^t$$

$$\boldsymbol{y}^t = sigmoid(\boldsymbol{o}^t)$$

3. METHODOLOGY

In this section, we discuss the gathering of training sets and testing sets. We also discuss the model used and flow of the training and testing of classifier for sentiment analysis.

Dataset Gathering. Since there's no publicly available Filipino tweet corpus, we made our own dataset. The tweets data were gathered using a Python script built with NLTK [13] library that uses Twitter Streaming API. The script searches for tweets that has smileys :) , :(, :D , :[, xD. For each tweet, the Twitter API sends a JSON file. The script writes the received JSON file to a new line. The tweets are then labeled as follows:

Table 1: Labels according to smileys

Label	Tweets with smileys
Positive (1)	:) :D xD
Negative (0)	:(:[

wherein positive tweets are labeled as 1 and negative tweets are labeled as 0. These labels conform with the output activation function sigmoid. Preprocessing of tweet data removed the irrelevant information, leaving only the main tweet text. Further cleaning of tweet texts was done to remove unnatural features and words like reserved words, hashtags, mentions, URLs, emojis and smileys. The whole dataset was then built with 92% (3177 tweets) as training set and 8% (256 tweets) as test set.

Model for Sentiment Analysis. Two types of RNN were considered for the model of sentiment analysis, the Simple RNN and the LSTM. The models were implemented as discussed in Section 2 with a slight variation of adding a Dropout layer. Dropout values of 0.1, 0.2 and 0.5, 0.8 were manually tested and it was found that 0.8 gives the highest accuracy, therefore a Dropout value of 0.8 were used in other variations of RNN/LSTM, as shown in Table 2. After the RNN/LSTM layer, a fully-connected layer is added with sigmoid as the activation function.

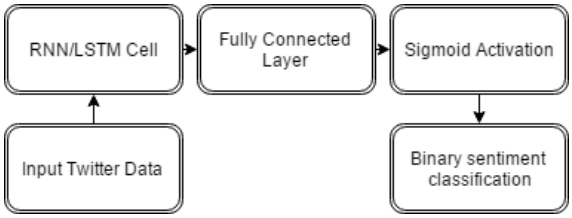

Fig. 3. Model of Sentiment Analysis

We trained the models at different number of hidden units as shown in Table 2, for 5 epochs. After 5 epochs, the model is already oscillating at 97% training accuracy. The test accuracy suffers overfitting for epochs beyond 5 epochs. Chollet et al. [2] said that the choice of batch size is tricky for RNNs, therefore we also trained the model in 2 values of batch size, 32 and 64. We used ADAM with learning rate of 0.001 as optimizer. Listing 1 shows the Keras code in Python for the case when LSTM is used.

4. RESULTS AND DISCUSSION

1698 positive tweets and 2410 negative tweets were used to train the model and 128 tweets each of positive and negative were used to evaluate the model. The combined dataset is 4108 tweets in size. The model only accepts numbers or sequences of numbers, therefore, before the training, a vocabulary of words that are contained in dataset were made. Each unique word has a numerical index which will be used to transform text to sequences of numbers. Therefore, for each sample in training and testing, the inputs are numbers, integers in fact which represent certain words. Also, the model can't use sequences of different length, therefore before the sequences are fed to the model, it is padded with zeroes until its length equals the specified length of the input of the model. The loss function used to compute the loss between the output and the target label is binary cross entropy as there are only 2 classifications for the model. The loss function $L(\theta)$ is

$$\boldsymbol{L}(\boldsymbol{\theta}) = ln\left(1 + \frac{\boldsymbol{y}^t}{1 - \boldsymbol{y}^t}\right) - \boldsymbol{T}_l ln\left(\frac{\boldsymbol{y}^t}{1 - \boldsymbol{y}^t}\right)$$

where y^t is the value of output activation sigmoid and $\boldsymbol{T}_l$ is the target label of the sequence.

The deep learning approaches were then compared to the machine learning approaches as discussed in Section 1.1. The codes for the related works in Section 1.1 are not available, therefore, we used an open-source implementation of Naïve Bayes found at [17].

Table 2: Sentiment Analysis percent accuracy as function of RNN type, number of hidden units and batch size and Models.

Model	Units	Batch Size	Negative	Positive	Mean
*sRNN	128	32	73.33	59.14	66.24
*sRNN	128	64	72.62	61.29	67.45
*sRNN	256	32	73.33	47.85	60.59
*sRNN	256	64	64.10	62.90	63.50
LSTM	64	32	71.28	67.74	69.51
LSTM	64	64	72.82	66.12	69.47
LSTM	128	32	78.46	61.82	70.15
LSTM	128	64	77.44	63.98	70.70
LSTM	256	32	78.46	66.129	72.29
LSTM	256	64	78.97	63.44	71.21
LSTM	512	32	73.84	**73.12**	**73.48**
LSTM	512	64	81.02	60.75	70.89
NB BI-GRAM	-	-	83.59	60.75	72.44
NB TRI-GRAM	-	-	**84.62**	58.60	71.92

***sRNN mean SimpleRNN or Vanilla RNN**

We observe from Table 2 that the accuracy for determining negative tweets peaks at the machine learning approach Naïve Bayes – SVM Tri-GRAM. For the positive tweets, the model peaks accuracy at LSTM with 512 hidden units and batch size of 32. It was also found that the model LSTM with 512 hidden units and batch size of 32 is the sweet spot between negative accuracy and positive accuracy. It also gave the highest mean accuracy between negative accuracy and positive accuracy. It is apparent in the table that most of the models have higher negative accuracy than positive accuracy. This could be caused by the imbalanced ratio of positive tweets and negative tweets in training set. It was also shown in Table 2 that models using simple RNN produce a significantly lower accuracy as compared to the models using LSTM, however Simple RNN beats LSTM at training time. Simple RNN trains twice as fast as LSTM does. LSTM therefore introduces a tradeoff between the performance of the model and the computational power required to train it with.

We also observe that the deep learning models have less difference between negative and positive accuracy than the machine learning models have. This shows that RNNs are more resilient to imbalanced dataset than Naïve Bayes approach.

Examples of negative tweets that were classified by the network as positive:

1. Next time, wag kayo magharrytan para walang masaktan
2. Aww sad ending bongga naiyak ako ng sobra pwede na pang watty toh
3. Grabi and sad ng ending

Examples of positive tweets the were classified by the network as negative:

1. Yey!! Pinansin nya ako…!! Hahaha.
2. Get ready guys . May twitter party tayo. Kelangan naten magtrend
3. Gusto ko madami kasi nga na-miss kita! Pag pasado ka na, balik ka na ah?

5. CONCLUSION

In this paper, we introduced a deep learning approach to sentiment analysis of Filipino tweets that accurately classify tweets into positive or negative. We used smileys as labeling feature for tweets. We found that certain combinations of model parameters achieve significantly higher performance in classification. We showed that LSTM significantly outperforms simple RNN architecture. We showed that certain RNN models outperforms Naïve Bayes approach although not by much margin. We also showed that RNN models are more resilient to imbalanced dataset than Naïve Bayes approach.

Previous versions of this paper used less number but balanced dataset and got balanced results between negative and positive accuracies. Therefore, in the future we would like to keep the balance in numbers between different classifications by using larger and balanced dataset.

In the future, we also would like to expand the number of classifications for sentiment analysis of Filipino tweets into more specific emotions. A more meaningful analysis could be done by correcting misspellings in tweets during the text preprocessing stage. Also, instead of using smileys as the labeling feature, the dataset should be labeled manually by humans. Use of CNN architecture for sentiment analysis of Filipino tweets is also a promising direction. Also, a standardized Filipino tweet corpus should be made to benchmark the performance of sentiment analysis models using different approaches like Naïve Bayes, SVM, and our proposed approach, Recurrent Neural Network.

Listing 1. Keras Code for the Model using LSTM

```
model = Sequential()
model.add(Embedding(max_features, 128))
model.add(LSTM(512, dropout=0.8))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])

# The codes are available at [16]
```

REFERENCE

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). http://tensorflow.org/ Software available from tensorflow.org.

[2] François Chollet et al. 2015. Keras. Github. https://github.com/fchollet/keras

[3] The Importance of Sentiment Analysis in Social Media. 2015. Available: https://results2day.com.au/social-media-sentiment-analysis-2/

[4] Michael Pippin Jr., Ron Odasco, Ronald de Jesus, Miguel Tolentino and Rex Bringula. 2015. Classifications of Emotion Expressed by Filipino through Tweets. International MultiConference of Engineers and Computer Scientists 2015 Vol. 1. IMECS.

[5] Jason Rennie, Lawrence Shih, Jaime Teevan and David Karger. 2003. Tackling the Poor Assumptions of Naïve Bayes Text Classifiers. In Proceedings of the Twentieth International Conference on Machine Learning 616 – 623. Available: https://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf

[6] Venkata Pagolu, Kamal Challa, Ganapati Panda and Babita Majhi. 2016. Sentiment Analysis of Twitter Data for Predicting Stock Market Movement. In Proceedings of the International Conference on Signal Processing, Communication, Power and Embedded Systems. *arXiv preprint arXiv:1610.09225*, 2016

[7] Fredetick Patacsil, Alvin Malicdem and Proceso Fernandez. 2015. Estimating Filipino ISPs Customer Satisfaction Using Sentiment Analysis. Horizon Research Publishing

[8] Jan Deriu, . 2016. Sentiment Analyis of Deep Convolutional Neural Networks with Distant Supervision. Swiss Federal Institute of Technology Zurich. Available: http://e-collection.library.ethz.ch/eserv/eth:49518/eth-49518-01.pdf

[9] Pengfei Liu, Xipeng Qiu and Xuanjing Huang. 2016. Recurrent Neural Network for Text Classification with Multi-Task Learning. In Proceedings of the Twenty-Fifth International Join Conference on Artificial Intelligence. IJCAI-16. Available: https://www.ijcai.org/Proceedings/16/Papers/408.pdf

[10] Rion Martin. Understanding Sentiment Analysis and Sentiment Accuracy. 2014. Infegy. Available: http://blog.infegy.com/understanding-sentiment-analysis-and-sentiment-accuracy

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press.. https://www.deeplearningbook.org.

[12] Denny Britz. 2015. Recurrent Neural Networks Tutorial, Part 1 – Introductions to RNNs. Available: http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns

[13] Bird, Steven, Edward Loper and Ewan Klein. 2009. Natural Language Processing with Python. O'Reilly Media Inc

[14] Chris Olah. 2015. Understanding LSTM Network. Available: http://colah.github.io/posts/2015-08-Understanding

[15] Hochreiter, Sepp and Schmidhuber, Ju ̈rgen. Long Short- Term Memory. Neural Computation, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco. 1997.9.8.1735. URL http://www.bioinf.jku. at/publications/older/2604.pdf

[16] https://github.com/iamRusty/Sentiment-Analysis-of-Filipino-Tweets

[17] Gregoire Mesnil. Naïve Bayes SVM (NB-SVM). Available: https://github.com/mesnilgr/nbsvm at/publications/older/2604.pdf