***Sentiment Analysis – Mapping from Sommelier comments to number ratings***

**W266 Project Progress Report**

**Summer 2018**

**Taiwo Raphael Alabi, Vishal Agarwal, Yisang Yoon**

***Abstract***

Sentiment analysis is one of the several areas of intense research in Natural Language Processing. The goal is to be able to map the summary meanings within a sentence or amongst a group of sentences to a label or group of labels. We investigate in this group project a sentiment analysis approach to classifying Sommelier sentiments on wines from various regions around the world and mapping these sentiments to a continuous numerical variable (wine score/wine grade). For sentiment analysis and mapping to a continuous domain, we investigated several state-of-the-art neural networks including: convolutional neural networks *(CNN)*, long-short term memory *(LSTM)* neural networks and gated recurrent unit *(GRU)* neural network. These neural networks were evaluated against each other for computational performance and perplexity scores. This project also explored using POS tags to extract features of most significant relevance to the eventual tag. These features include Sommelier comments with either a noun and/or adjective part of speech. These enriched wording were then explored for use as word-embeddings to feed into the various neural networks described above. Results in the discussion section also show how loss and mean-absolute error can be improved in a GRU via changing the number of epochs, number of hidden units and word-embedding vector size.

We were able to successfully match Sommelier sentiments to ratings using all 3 investigated neural networks. GRU was found to be more computationally efficient with a mean absolute error value of 0.0244.

***Introduction and Background***

The goal of this NLP project is to use the output from Master Sommelier’s on over 60,000 wines as training data to fit to labels provided by same Sommeliers. The labels are continuous data on a 100-point scale. The goal is to generate an *LSTM*, *GRU* or *CNN* algorithm that can map the descriptive wine taste (from an expert Sommelier) to a point scale. A further extension of the project is to concatenate the wine-type and wine-farm-location into the description (enable these to have word-embeddings that contribute to the eventual word embedding matrix) and to evaluate the accuracy improvement from including these features. A generative model will then be built where given a wine with some percentage point value and source the model can map this to a domain space containing all adjectives that potentially could be used to describe the wine.

Sentiment analysis is currently a burgeoning research field in linguistics. There have been multiple research papers since the 2000’s that have focused on improving sentiment analysis accuracy for various use case scenarios. [2][3][4] One of the key potential use of sentiment analysis is for e-commerce and e-trade. Sentiments(comment(s) and complaint(s)) left behind by potential customers or loyal-customers can be mined to better understand the general direction businesses should take to retain or attract more customers. [9][10][11]

In the literature multiple approaches towards sentiment analysis have focused on the use of state-of-the-art neural network techniques including *CNN*, *RNN*, *bi-LSTM*, *LSTM* and *GRUs*.[6][8][9][10][11][12] Some other techniques have also involved the combination of anyone of the above technique for generating sentiment analysis with accuracies up-to and exceeding the 90% percentile. While most techniques in the literature have focused on mapping sentiments to categorical (binary to ternary) variables, there has been only a handful of research that has dealt with mapping sentiments to a continuous variable.

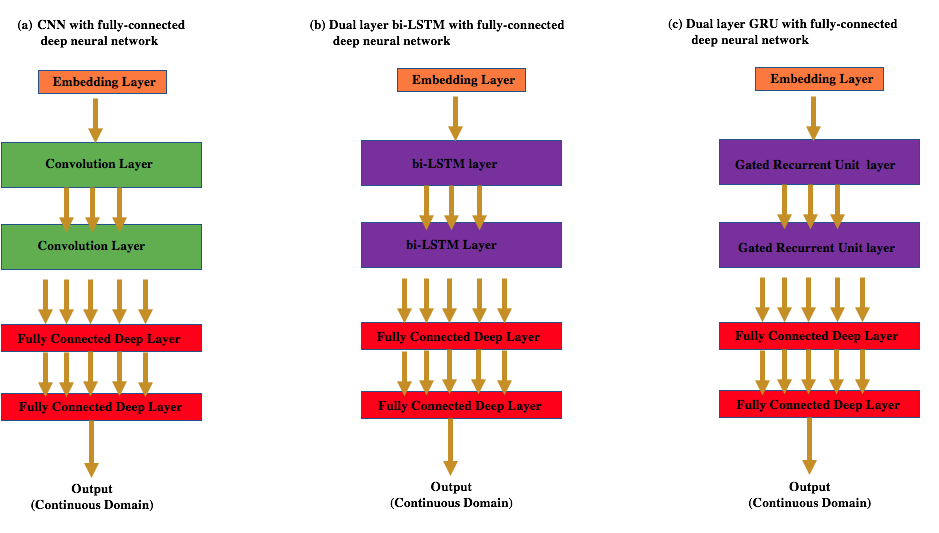
This research project explores mapping of sentiments to a continuous variable domain (in percentage points). Model improvement accuracy was investigated using a custom nltk POS tagger for identifying specific features(words) within the sentiment sentence space. By using POS tagged sentiments, we extracted highly relevant features from the sentences that we believe could improve initial model accuracy via enrichment of the feature space with word-vectors that more accurately reflect the overall sentiment. The dataset used for this analysis was downloaded from Kaggle and is available as .csv and as .json files.

Wine tasting experts, Sommeliers, can be a great resource/recourse for recommending wines for special events and also for providing general knowledge about wines and their tastes. Even though people without formal training can claim to be wine-tasting experts, to become a Sommelier one has to undergo years of both formal education, professional training and also build experiences year over year. The cost to become an expert, certified Master Sommelier is very expensive. As of the end of 2013, there are only 214 certified Master Sommelier’s in the world[1]. This underscores the importance of using sentiment analysis of comments from master Sommeliers as a template for predicting scores for wines. The reverse can also be viewed as an interesting concept: predicting expert Sommelier comments from the score rating of a wine.

***Methods***

The main dataset used for this exercise is the winemag dataset from Kaggle, available at <https://www.kaggle.com/zynicide/wine-reviews>.The dataset contains the reviews from wine tasters on a variety of wines and the points they have assigned to the wines based on the taste-review.

Three neural network models were investigated for this sentiment analysis: They include a *CNN* model, *bi-LSTM* model and a *GRU* model. The architectures of the models are shown below.

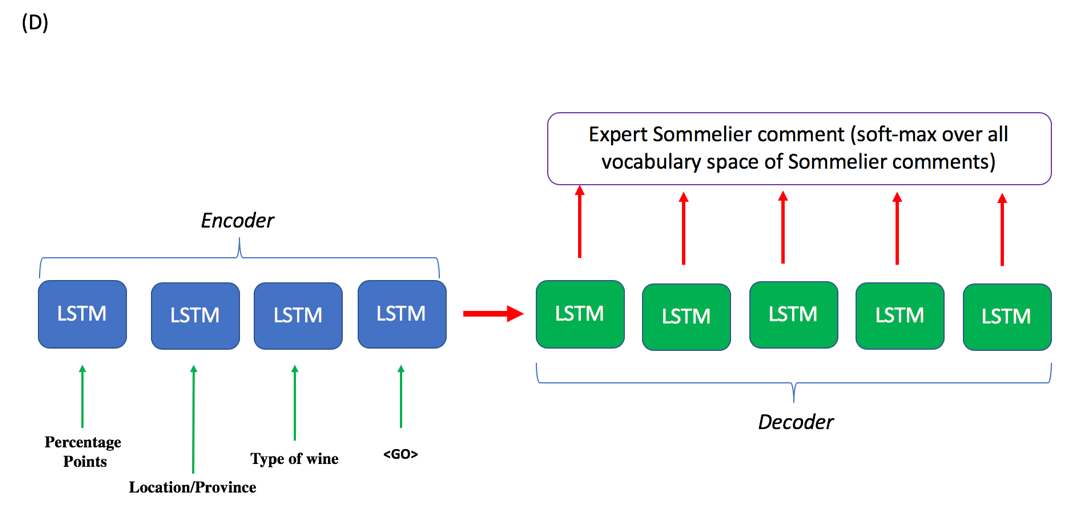


The embedding layer for all the models investigated used an embedding size of 500. Instead of using pre-trained word-embeddings from Glove, the model was allowed to train its own word embeddings. More details about data preparation are in the Data preparation and Results section.

The CNN architecture consists of two convolution layers and one global-max pooling layer. Dropout was implemented to help prevent overfitting. Regularization (*L1*/*L2*) was not implemented which may account for the overfitting-like behavior shown in the results from the *CNN* sentiment analysis in the result and discussion section. A 2-layer deep neural network was used to map the max-pooled layer to a continuous domain. The network for trained for 10 epochs and a stochastic gradient descent optimizer (SGD) was used with a learning rate of 0.01.

The bi-LSTM architecture investigated consists of two bi-directional LSTMs and two fully-connected deep neural network. Spatial drop-out was implemented on the bi-directional LSTM as well as L2 regularization to prevent overfitting. The hidden layer size of the bi-directional LSTM was chosen as 500. The bi-directional LSTM was trained for 10 epochs with an sgd. optimizer and a learning rate of 0.001.

The GRU architecture also consisted of two GRU’s with an hidden layer-size of 500 connected to a two-layer deep neural network. Regularization was not implemented, but drop-out was implemented to prevent overfitting to the train dataset. The GRU was trained for 10 epochs with an sgd. optimizer and a learning rate of 0.001.



The seq-2-seq encoder-decoder algorithm architecture used is also shown above. The input to the LSTM for encoding were the percentage points (pre-processed for conversion to an A, B, C scale since integer values have no word-embedding representations), location/province, type of wine and a pre-tag *<GO>*. The decoder was trained with inputs appended with an *<EOS>* tag. Both the decoder and encoder LSTMs had a hidden state size of 500. The input and output sentences were padded to be of the same length before they were fed into the Encoder and Decoder LSTM.

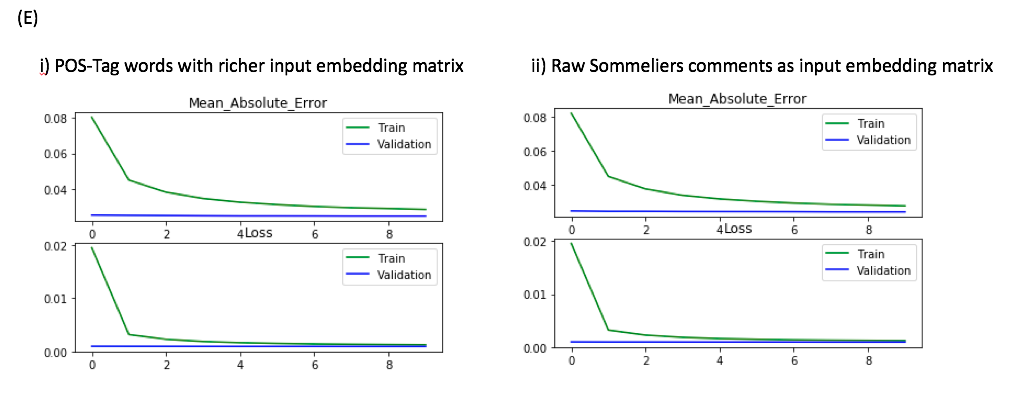
A note here that attention mechanism was not evaluated in this section due to time constraint. This architecture with an attention mechanism or only attention mechanism by itself, based on available literature, is expected to perform better at such sequence to sequence translation problem.[14][15][16]

***Results and Discussion***

*Data-Preprocessing:*

The percentage points (with range from 80-100%) were normalized by dividing by 100 to give a range between 0.8 and 1. Normalization amongst other benefits also ensures that we do not see the exploding gradient issue when running LSTMs.

In addition, because the word-embedding space of the Sommelier descriptions is ripe with words that carry non-essential meanings like articles, determiners, punctuations etc., we explored using a sentiment space that was reformatted. Each Sommelier comment was passed through a POS tag algorithm (NLTK pos-tagger). The words with POS tags having *‘JJ’, ‘JJR’, ‘JJS’,* *‘NN’, ‘NNS’, ‘NNP’, ‘NNPS’*, and *‘VBN.’* were used as a new feature space and served as input to the neural architectures previously shown above. The graph below shows the comparison between the loss and mean-absolute error from a GRU architecture fed with raw Sommelier comments vs a GRU architecture fed with POS tagged words carrying a richer embed matrix.



The loss and mean-absolute error values for both models do not appear significantly different, however there appears to be a very slight improvement in the initial loss function score for the pre-formatted comments using POS tags. This initial gain in loss and mean absolute error is carried across through the entire training and is reflected in the overall loss and mae values at the end of the training. The mean absolute error and raw-different for GRU trained with raw Sommelier comments are 0.0245 and -0.0037 on the validation dataset vs 0.0244 and -0.0034 on GRU trained with POS tagged words.

All the other model results discussed below were evaluated using a word-embedding space pre-formatted with POS tags.

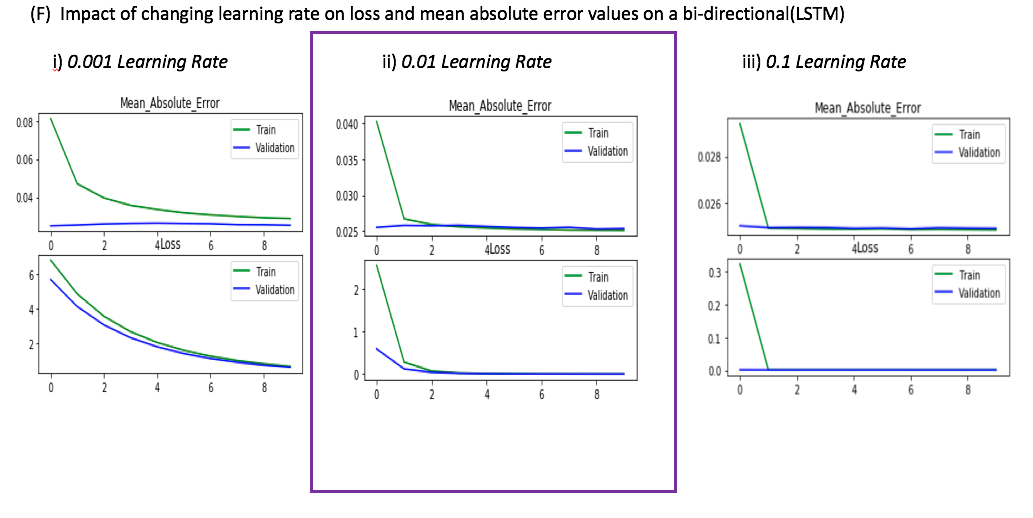
Data pre-processing for the seq-2-seq model was done for both the decoder and the encoder input matrix. The input to the encoder consisted of the a <GO> pre-tag, percentage points reformatted to a scale of tags between *AA (>0.95)*, *BB (0.9<BB<=0.95)*, *CC (0.85<CC<=0.9)* and *DD (<=0.85)*; country, province and wine variety.

The input to the decoder LSTM consists of word embeddings from the raw Sommelier description accompanied with an <EOS> tag. Both input to the Encoder and Decoder LSTMs were padded to the same length before they were fed to the embedding layer.

Tuning of two hyper-parameters were investigated on this project- The number of epochs and the learning rate of the SGD optimizer. The number of epochs of training was varied from 5 epochs to 10 epochs for each model. As evident in the results section, 5 epoch(s) are not sufficient enough for training the GRU and the LSTM model since the MAE as well as loss are still been optimized. However, for the CNN algorithm training for 5 epochs vs training for a longer period does not show a very significant change in the MAE or loss.

To ensure uniformity across all training models we decided to stay with a training epoch of 10 for all models investigated.

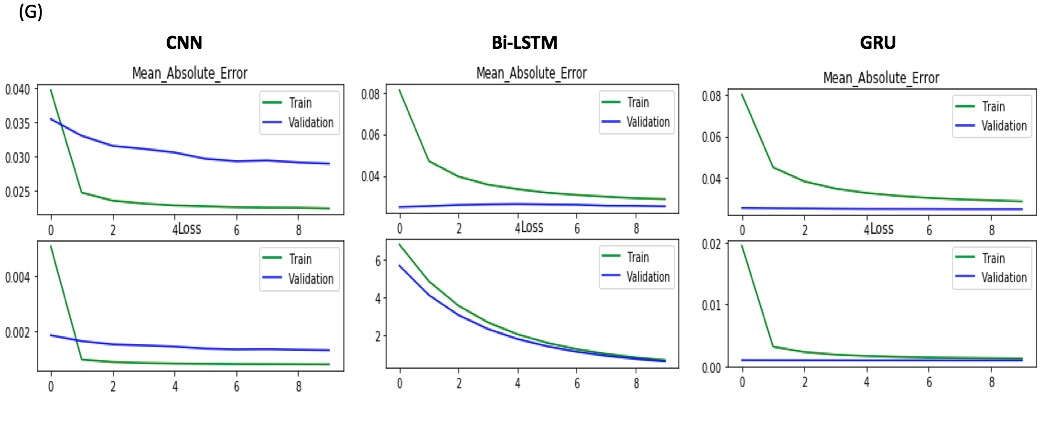
We went ahead with using a learning rate of 0.001 across all architectures primarily because the CNN network appeared to already be overfitting at the current learning rate of 0.001. Although it does appear going with a learning rate of 0.01, as shown below for the bi-LSTM, may have better.



*Results:*

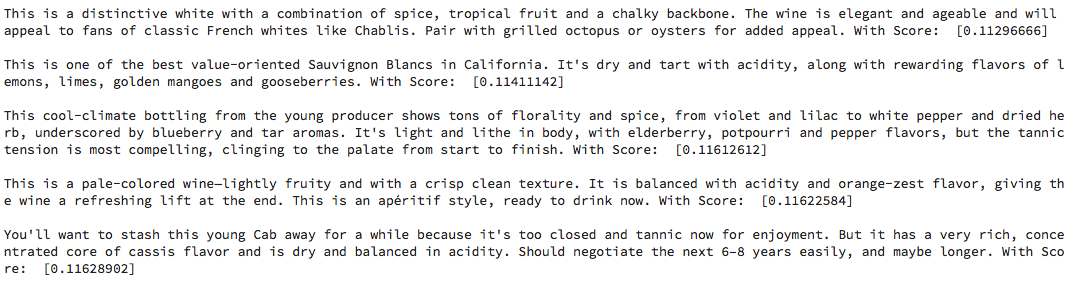
From the results shown below (note the y-axis of all 3 graphs are not on the same scale), the CNN algorithm appears to perform worse on the validation dataset when compared to bidirectional LSTM and the GRU neural architectures. On the CNN architecture there was no regularization used although dropout was used to help prevent overfitting. The GRU architecture, however, also did not use regularization but the overall performance on the validation data outperforms that of the CNN and slightly better than the LSTM algorithm.

It is interesting to note that loss for a bi-LSTM starts initially at a high value but upon training drops much faster and uniformly on both the training and the validation dataset than it does on the GRU and the bi-directional LSTM. The GRU architecture also seems to perform slightly better than the CNN and the bi-LSTM algorithm in terms of the loss.



It should be noted that loss and mean-absolute-error decay over time on both the GRU and bi-LSTM is indicative that both algorithms will benefit with more training and will potentially outperform the CNN algorithm in mean-absolute error as well as loss. The GRU algorithm, in terms of train time, is also superior to both the bi-LSTM and the CNN as it takes a relatively shorter train time. This is expected considering that there are more trainable parameters in the CNN network- 10.7 million (1st layer convolution)- compared to 1.5million (1st layer GRU) trainable parameters on the GRU architecture.

Of the 3 models investigated, the GRU is the model for the same train period, batch-size and epoch; the GRU delivers on low perplexity scores and low mean absolute error. After training the GRU for 4 iterations a few lines of codes were written to extract features that have the maximum deviation between the absolute value of the (y\_pred – y\_test). The features(descriptions) that produced the maximum absolute deviations are shown below.



When the sentences above are compared against sentences that have lower absolute error values it does appear that the more the wine is described in a storyline format (meaning described in comparison to something else) the more confused the algorithm gets in trying to predict the final rating. To eliminate such confusion I think datasets with similar ‘synonym-like’ rating will need to be fed to the GRU algorithm.

A seq-2-seq translator algorithm was also investigated for use in mapping wine rating, source, variety to a predicted Sommelier comment space. Training of the seq-2-seq generative network was done on only a subset(20k) of the 120k dataset because training takes an immensely long time. After training the network for 4 epochs perplexity as low as 1.04 was obtained on the trained dataset and a perplexity of 1 was obtained on the test dataset. It should be noted that these perplexity values are likely to change when the algorithm is exposed to the full dataset of 120k. The addition of an attention algorithm that is state-of-the-art for sequence-2-sequence translation is also expected to significantly improve model accuracy when the algorithm is eventually exposed to the full dataset.[14][15][16]

***Conclusion***

In summary we have shown that Sommelier’s sentiments can be successfully mapped to a continuous domain using one of several state-of-the-art neural networks including CNN, LSTM and GRUs. We have shown that the computationally faster and more efficient approach is using GRUs. This has been attributed to GRU’s having less trainable parameters when compared to either the bi-LSTM or the CNN. We have also shown a slight improvement in initial loss when POS-tagged words including Nouns and Adjectives were used as input to the neural network than when raw-inputs containing determiners etc. were used. We believe this is a result of the enrichment of the word-vector space with words carrying more significance from POS tagging.

*Link to codes:* [*https://github.com/taiworaph/Natural\_Language-\_Processing*](https://github.com/taiworaph/Natural_Language-_Processing)

***References***

[1] <https://www.winebags.com/Sommeliers-The-Wine-Experts-s/1887.htm>

[2] Sundermeyer M, Schlüter R, Ney H. LSTM Neural Networks for Language Modeling.

[3] Gal, Y. and Ghahramani, Z., 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems* (pp. 1019-1027).

[4] Stojanovski, D., Strezoski, G., Madjarov, G. and Dimitrovski, I., 2016. Finki at SemEval-2016 Task 4: Deep learning architecture for Twitter sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (pp. 149-154).

[5] Wang, J., Yu, L.C., Lai, K.R. and Zhang, X., 2016. Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Vol. 2, pp. 225-230).

[6] Kim, Y., 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[7] Goldwater, S. and Griffiths, T., 2007. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th annual meeting of the association of computational linguistics* (pp. 744-751).

[8] Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: capturing favorability using natural language processing. *In Proceedings of the 2nd international conference on Knowledge capture (K-CAP '03).*

[9] Cliche M. BB\_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. arXiv preprint arXiv:1704.06125. 2017 Apr 20.

[10] Cambria, E., Poria, S., Gelbukh, A. and Thelwall, M., 2017. Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, *32*(6), pp.74-80.

[11] Arras, L., Montavon, G., Müller, K.R. and Samek, W., 2017. Explaining recurrent neural network predictions in sentiment analysis. *arXiv preprint arXiv:1706.07206*.

[12] Conneau, A., Schwenk, H., Barrault, L. and Lecun, Y., 2016. Very deep convolutional networks for natural language processing. *arXiv preprint*.

[13] Hassan, A. and Mahmood, A., 2017, April. Deep learning approach for sentiment analysis of short texts. In *Control, Automation and Robotics (ICCAR), 2017 3rd International Conference on* (pp. 705-710). IEEE.

[14] Cho, K., Merrienboer, Bart., bahdanau D., Bengion Y., On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. 2014arXiv1409.1259C

[15] Luong, M.T., Pham, H. and Manning, C.D., 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

[16] Madhyastha, P., Neural Machine Translation.