

Predicting Steam Game Reviews Using Recurrent Neural Networks

Daniel Lee
School of Computing
Queen's University
Kingston, Canada
18dil@queensu.ca

I. MODEL

Motivation: In this project, I want to provide a deep learning solution to accurately predict Steam game reviews as positive or negative (i.e., “Recommended” or “Not Recommended”). The Steam Community utilizes “Recommended” and “Not Recommended” ratings to understand the overall feelings about a game. Traditional techniques such as sentiment analysis are not always accurate, due to the lack of context, sentiment ambiguity, sarcasm, comparatives, regional differences and degree of agreement between humans and machines (i.e., human concordance) [2]. Since, gamers tend to use sarcastic language it makes sentiment analysis challenging to be accurate.

Approach: I will use a variable-length word vector as my input using the text of Steam game reviews. I explain the general pre-processing steps in Section II. Thus, I will be using a word embedding network such as *Word2Vec* to generate the word vectors.

By using *Python* with the *Keras API*, I will build a 3-layered recurrent neural network (i.e., RNN) with 2 dense layers, and 2 Long-Short Term Memory networks (i.e., LSTM). I chose a recurrent neural network for my text classification problem because it is robust and known to perform well with variable-length text [1], [4]. Thus, my network architecture will be based on a recurrent neural network.

Furthermore, the output will be a binary sentiment of either 0 or 1, where we are trying to classify between “Recommended” and “Not Recommended”. More details on the binary classification are in Section II.

II. EXPERIMENT

In my project, I will be using Steam game reviews from the Steam distribution platform as my primary dataset. We have crawled 91,321 Steam game reviews that are labelled as 1 or 0 (i.e., “Recommended” or “Not Recommended”). We chose the Steam distribution platform because it is one of the largest digital distribution platforms for PC gaming [3].

To prepare the input for the recurrent neural network, we will use 80% of the Steam game review dataset for training, and set aside 20% for testing. We will then pre-process the training set by the following:

- 1) Tokenization

- 2) Punctuation removal
- 3) Hyperlink removal
- 4) Stop-word removal
- 5) Stemming

With respect to the hyper-parameters of the recurrent neural network, I will utilize a sigmoid activation function for the output layer with a binary cross-entropy as my loss function because it is a binary classification problem. In addition, I will use dropout, which is a regularization method, to prevent overfitting of my data.

Furthermore, if the time permits, I want to compare the two approaches: my deep learning approach, and the existing out-of-box sentiment analysis models (e.g., Google, Amazon, IBM) with respect to the precision, recall, F1 score, and AUC.

REFERENCES

- [1] G. Arevian, “Recurrent neural networks for robust real-world text classification,” in *IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society, 2007, pp. 326–329.
- [2] B. Donkor, “Sentiment Analysis: Why It’s Never 100% Accurate,” <https://bmrd.me/posts/sentiment-analysis-never-accurate>, 2014, (last visited: February 11, 2019).
- [3] D. Lin, C.-P. Bezemer, Y. Zou, and A. E. Hassan, “An empirical study of game reviews on the steam platform,” *Empirical Software Engineering*, pp. 1–38, 2018.
- [4] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, “Text classification improved by integrating bidirectional lstm with two-dimensional max pooling,” *arXiv preprint arXiv:1611.06639*, 2016.