Lyric-based approach for music emotion recognition using hierarchical attention networks

Tri-Nhan Do
Advanced Program in Computer Science
Faculty of Information Technology
University of Science, VNU-HCM
dtnhan@apcs.vn

Tan H. Nguyen
Advanced Program in Computer Science
Faculty of Information Technology
University of Science, VNU-HCM
nhtan@apcs.vn

Tri M. Nguyen
Advanced Program in Computer Science
Faculty of Information Technology
University of Science, VNU-HCM
nmtri17@apcs.vn

***Abstract*—Music emotion is one of the most important features in music recommendation. To enhance the quality of music recommendation, we propose a method to classify the emotion of songs by using recurrent neural network. We utilize the natural structure of a song which is words combine to lines, lines combine to segments, and segments combine to a complete song by adapting a hierarchical attention networks (HAN). We test the model on the dataset that have classified songs as positive or negative emotion. Our result does not improve so much when compare to the state-of-the-art model but the improvement of HAN is we can visualize how the model giving the prediction by ranking the important of lines and words in the song. It help us provides the link between a computational perspective and natural language perspective that differentiate music emotion.**

I. Introduction

The number of people listening to online music websites is increasing dramatically accompanied with services to improve user experiences. One of the most significant services is music recommendation system. These recommendation systems filter information to predict users' preference of a certain song.

There are two main approaches for music recommender systems, which are Collaborative Filtering and Content Based Filtering [1]. Collaborative Filtering approach bases on similarity between users' behaviours, activities or preferences. The major problem of this approach is that it requires a large dataset to achieve high accuracy, which can lead to a "cold start" problem. In contrast, Content Based Filtering is the most common techniques for music recommender system. In this approach, the model closely analyzes the characteristics of a song to make a recommendation. This leads to the demand for classifying a song.

According to proposed methods for music classification, music can be classified according to artist, genre, instrument or mood of the song [2]. However, classification based on songs' mood is the most interesting and challenging approach. [3] The approach to classify music varies in different ways. One of a potential one is to use subjective human feedbacks or user tags [4]. However, a drawback of this method is that it is hard to collect dataset since users are not always willing to provide their feedbacks. Of several hundred users surveyed, listeners indicated that vocals (29.7%), lyrics (55.6%), or both (16.1%) are among the most salient attributes that they notice in music [5]. Another approach is to classify based on its characteristic like audio and lyrics [6] [7] [8] [9]. However, not only does the song's characteristic but also feeling of different people based on the situations that it is played makes it difficult to evaluate the mood of the song correctly.

There are several researches on this topic and standard ground truth for it has been improved day by day such as MIREX [10]. Although dataset collected by MIREX has high reputation and is created by many experts, it is not reachable outside the competition. Last.fm, is another reliable dataset, which uses valence and arousal values of the word based on Russell's model, this dataset is bigger than most of the current publicly available datasets [11]. The dataset we used was created by Çano (2018) in his thesis named MoodyLyricsPN. It contains 5,000 songs with 2,500 songs were labeled as positive and others were labeled as negative [12]. There are also others hand-labeled datasets which are created for private projects or they are generally lack of content-based retrieval methods. Some datasets use machine learning techniques instead of human experts to extract emotions in Music. [13].

To determine mood of a song, some studies use lexicon based, whereas others apply machine learning approaches [14] [15] [16]. Our proposed method is cleaning the text for feature pre-processing, word embedding to represent words as dense vectors and using Hierarchical Attention Networks for training. We have experimented some pre-trained word vector extraction methods like fastText, word2vec, GloVe to choose the most proper feature extraction. To validate the quality of the predicted mood, we compared our model with the result of Çano on his dataset. The evaluation process reveals an accuracy of 76%, which is slightly improvement from Çano's result [12].

The rest of the paper is organized as follow: Section 2 reveals some related works on sentiment analysis and music emotion classification, section 3 introduces the authors' proposed method using lyrics analysis, section 4 presents the results and discusses some future works.

II. Related work

The approach of us is to analyze mood of a song based on its lyrics. Therefore, the problem of sentiment analysis from textual is the main focus of this paper. There are many works that were conducted in sentiment analysis field. Some of them apply machine learning method or lexicon-based approach separately, whereas others combine both of the two methods together. Some of the most significant related works are listed below.

A. Twitter sentiment analysis

Lexicon-based method and machine learning approaches are applied to analyze the variation of the public opinion about retail brands [17] . By using semantic score assigned to each words, lexicon-based method can estimate the variation of a tweet and also include tags in the speech. Regarding the machine learning approach, by employing Naive Bayes and Support Vector Machines classifier, this method overcomes the problem of excessive dependence on words from the dictionary. The main contribution of the project is combining the two approaches together by extracting features from lexicon score for Naive Bayes and SVM classifier.

B. Music mood classification using machine learning

A recommendation system was built based on a Naive Bayes classifier [18]. By analyzing lyrics of songs, the method classifies training dataset containing 1000 songs into 2 moods, happy and sad. The most significant contribution of this project is the creation of dataset which was filtered, labelled, and publicized. Moreover, the Naive Bayes in this project yields the result of 72.5%.

C. Emotion Detection from textual source by using Natural Language Processing

In emotion analysis, types of feelings are calculated based on any given text. This approach is classifying mood of a song based on English keywords denoted feelings like happy or sad [19]. Textual content from social networking site is obtained and defined into structure of list of sentences, list of tokens, word forms, word lemmas, and associated tags. After re-defining their structure and pre-processing data, the Naive Bayes approach is applied.

D. Simple and Practical lexicon based approach to Sentiment Analysis

The project analyzes sentiment of Twitter data by using lexicon based approach [20]. The manually created lexicon used in this method contains common or default Sentiment words, Negation words, Blind negation words, and Split words. Then Sentiment Calculation algorithm containing if-else functions is applied to aggregate the sentiment of the tweets.

E. Word Vector for Sentiment Analysis

This project proposes the approach of combining unsupervised and supervised method for capturing between semantic and sentiment similarities among words. The comparison between the proposed approach of word representation learning model and some other commonly used vector space models such as Latent Semantic Analysis, Latent Dirichlet Allocation, and Weighting Variants indicates that when capturing word representation instead of latent topics, the performance of the model improves.

F. Rule-based model for Sentiment Analysis of Social Media Text

To achieve high speed as well as extremely high accuracy in sentiment analysis on a large scale of dataset, a high quality lexicon is a crucial requirement. The paper presents gold standard lexicon, which is produced by combining qualitative and quantitative method and under consideration of grammatical and syntactical general rule for sentiment intensity expression and emphasis, especially attuning to microblog-like contexts. The evaluation indicates that when the lexicon is used as feature for VADER, a rule based model for sentiment analysis, the engine outperformed individual human rates and especially well in social media contexts. VADER sentiment lexicon is similar to LIWC, a commonly used lexicon in social media domain, which are both validated by humans.

III. Proposed method

First we will have a better look at the MoodyLyricsPN datasets. Next we will show to do pre-processing. Then we will introduce some feature extraction which we have mention above. Finally we will introduce the machine learning method that we will apply here.

A. ***Data Acquisition***

We will have some introduction about the MoodyLyricsPN dataset. The dataset is the list of songs that have been annotated using Last.fm user tags in one of the 2 categories: Positive and Negative [21]. The data contains a balanced version with 2500 songs for each of the 2 categories, totalling in 5000 songs [12]. We use the custom script to collect song lyrics from lyrics.wikia.com.

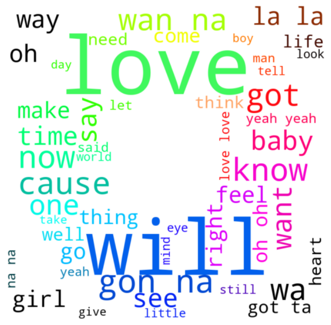

Fig. 1. WordCloud for Positive song

B. ***Pre-processing***

Before employing machine learning approach to extract sentiments, the typical pre-processing procedure is applied.

First we must remove all lyrics that has length too short or has a ”No Lyric” warning. Those are about 232 lyrics of these type, which reduce the number of sample from 5000 to 4768. Although we lost about 4.64% of total number of samples, we believe the remain amount still big enough to use.

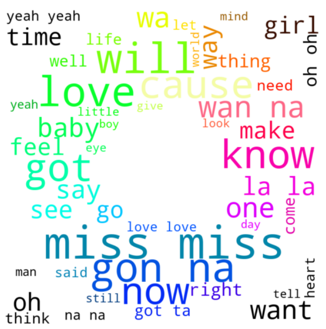

Fig. 2. WordCloud for Negative song

After lowercase all the remain lyrics, next we must remove all chorus and versus from songs lyrics. Those are prefix of sentence that tell us that how many time that sentence is told, for example: ”chorus(x3) Ha” mean we must repeat Ha three times. Although these thing can be important, we still remove them since they are not impact much to our model.

We then apply text normalization techniques, which is lemmatization from the NLTK library to achieve the root forms of inflected words. We choose WordNet Lemmatizer that uses the WordNet Database to lookup lemmas of words. Also, accroding to Erion Çano, we should remove stopword {“the”, “these”, “those”, “this”, “of”, “at”, “that”, “a”, “for”, “an”, “as”, “by”} so we do the same thing.

To handle lengthened words like humming, we apply exaggerated word shortening to simplify them. Words which have same letter more than two times and not present in the lexicon are reduced to the word with the repeating letter occurring just once. For example, the exaggerated word “NOOOOOO” is reduced to “NO”.

We use LabelEncoder to normalize labels and transform non-numerical labels Negative and Positive (as long as they are hashable and comparable) to numerical labels (encode labels with value 0 and 1 for two above class)

Finally, we divide the datasets into Train, Valid and Test. The Train has 3337 samples (70%), the Valid has 477 samples (10%) and the Test has 954 samples (20%). The result from the Test will be compare with the baseline.

C. Feature Extraction

Here we will use feature extraction like fastText, word2vec, GloVe and try to combine them with the model to gain result as good as possible.

*1) **fastText**:* : fastText is a library for learning of word embeddings and text classification created by Facebook’s AI Research (FAIR) lab. The model allows to create an unsupervised learning or supervised learning algorithm for obtaining vector representations for words. Facebook makes available pretrained models for 294 languages. fastText uses a neural network for word embedding. [22]

*2) **word2vec**:* : this is a common feature extraction in NLP. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space. [23]

*3) **Glove**:* : is an unsupervised learning algorithm developed by Stanford for generating word embeddings by aggregating global word-word co-occurrence matrix from a corpus. The resulting embeddings show interesting linear substructures of the word in vector space. [24]

Those are feature extraction we believe that if apply to our model it will gain optimal result.

D. Hierarchical Attention Network:

We follow the instruction of Yang et al. to create our model [25].Each structure level will have one layer of bidirectional gated recurrent unit (GRU) with attention applied to the output. The context vector is created by weighted sum the attention weights and then passed as the input to the next layer. The structure of the model can be seen in Figure 1, where there are the attention layers at word and sentence level. We will briefly introduce the various components of the model.

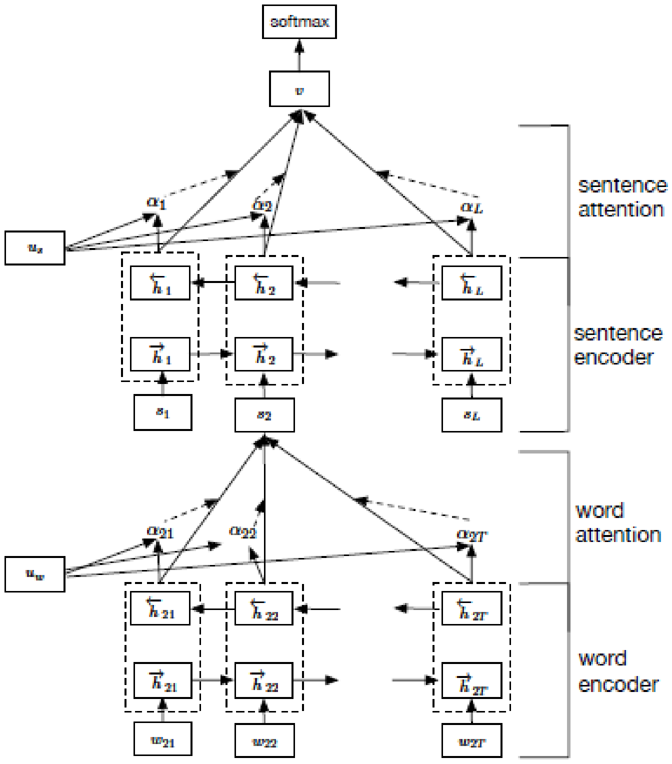

Fig. 3. Hierarchical Attention Networks

1) Word Embeddings: Featuring by Bag of word (BOW) based on frequency cannot express the connections between words, it discards word order thereby ignoring the context and in turn meaning of words in the song. Naïve Bayes Method is mainly used when the size of the training set is less. If categorical variable is not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. Therefore, in feature engineering step, we use prediction based embedding. The words that have the same meaning have a similar representation and closer together in related words coordinate system.

There are a wealth of way to approach word embedding, the most commonly hypothesis is "words that occur in the same contexts tend to have similar meanings" [15]. Word embeddings are a family of natural language processing techniques aiming at mapping semantic meaning into a geometric space. The word2vec is combination of two techniques – CBOW(Continuous bag of words) and Skip-gram model. These techniques learn weights from a large corpus of text to represent word as a vector in the space. The Global Vectors for Word Representation algorithm is an extension to the word2vec method for efficiently learning word vectors. GloVe constructs word co-occurrence matrix using statistics across the whole text corpus. FastText is is essentially an extension of word2vec model, but it treat character as the smallest unit to train on (character n-grams). It allows computing word representations for words that did not appear in the training data. we try to own train model by one-hot encoder, use GloVe pre-trained and finally decided to use GloVe with 300 dimensions to featuring vector because it gave better results.

Here is a full information about how we create embedding layer with pre-trained GloVe. We choose the GloVe with 6 billion tokens, 400,000 vocabulary size and uncased. Each word returns a vector with 300 elements, which means that word is represented in a spatial coordinate with 300 dimensions.

2) Gated Recurrent Units: GRUs, proposed by Chung et al. [26], uses gating mechanism to capture long-term dependencies in RNNs. There are two types of gate: the reset gate and the update gate. The update gate decides the important of the previous hidden state to the next hidden state which means that it controls how much information from the previous hidden state will be passed to the next hidden state. The reset gate helps control the amount of information of the previous hidden state that will be kept in the memory.

3) Attention Mechanism: Introduced by Bahdanau et ai. [27] to solve neural machine translation problem, attention have became famously worldwide due to its brilliant idea. Instead of storing all the information into one dense vector, the model will try to learn which words are important to achieve the objective. Like in our methods, we expected that the model will know which sentences and which words are important to determine the emotion of a song.

4) Classification: After obtained the song vector from attention layers, there is a final softmax layer to do the classification task where we just take the entry with highest probability as the prediction of the song. The cross-entropy loss over training set was used to train the model.

IV. Experiment

A. *Baseline Model:*

We compare the performance of HAN with the result reported by the author of MoodyLyricsPN dataset which is 75.63%

In addition, we also tested basic deep neutral network models such as CNN and LSTM on two pre-trained GloVe and Fasttext sets to compare with our method.

Here is a full information about how the authors create embedding layer with pre-trained fastText. We choose the fastText with 1 million words trained on Wikipedia in 2017, UMBC web base corpus and statmt.org news dataset. Each word returns a vector with 300 elements, which means that word is represented in a spatial coordinate with 300 dimensions. From 21391 words from 3150 songs of the train dataset is embedded with the corresponding word vector in fastText to build up 21391 300-dimension vectors. Then, each word in a song is replaced by its corresponding vector.

The final step is to train a classifier with Deep Neural Net-works. The we run model Convolutional Neural Network(CNN), Long Short Term Model (LSTM) from keras library. With CNN model we deploy4 layers: 1 embedding layers, 1 convolutional layer and 2 fully connected layer. For the convolutional layer, we use the length 3 of the 1D convolution window. All hidden layers are equipped with the rectification (RELUs) non-linearity. We use spatial drop out 1D after the embedding layer and dropout after the dense layer but not after the convolutional layer. From the song lyrics, we convert it to sequences of tokens and pad it to ensure equal length vectors of 250. Then we put it into the embedding layer before extract feature with convolutional layer because vectors is high dimensional and sparse. After the convolutional layer, we use global max pooling layer to feed directly feature maps into feature vectors. From feature vectors, we apply fully connected and sigmoid function to calculate probability that this song will have a positive or negative mood. Between this, we have set dropout rate as 0.25 for the dense layer. To rate our model, we use binary-cross entropy function as an objective function and Adam optimization algorithm to optimize our parameters.

With LSTM model, compare to CNN, this RNN model learns information from immediate previous step it can be updated while training the neural network. The embedding layer encodes the input sequence into a sequence of dense vectors of dimension, the LSTM transforms the vector sequence into a single vector, containing informa- tion about the entire sequence. The other hyper parameters like dropout, batch size are similar to that of CNN.

B. *Model Configuration:*

The lyrics had to have the uniform length to be the input for model. To identify the maximum sentences and maximum number of words in the sentence, we have plot the distribution of number of sentences in the song and the distribution of number of words in the sentence with Figure 2 and 3.

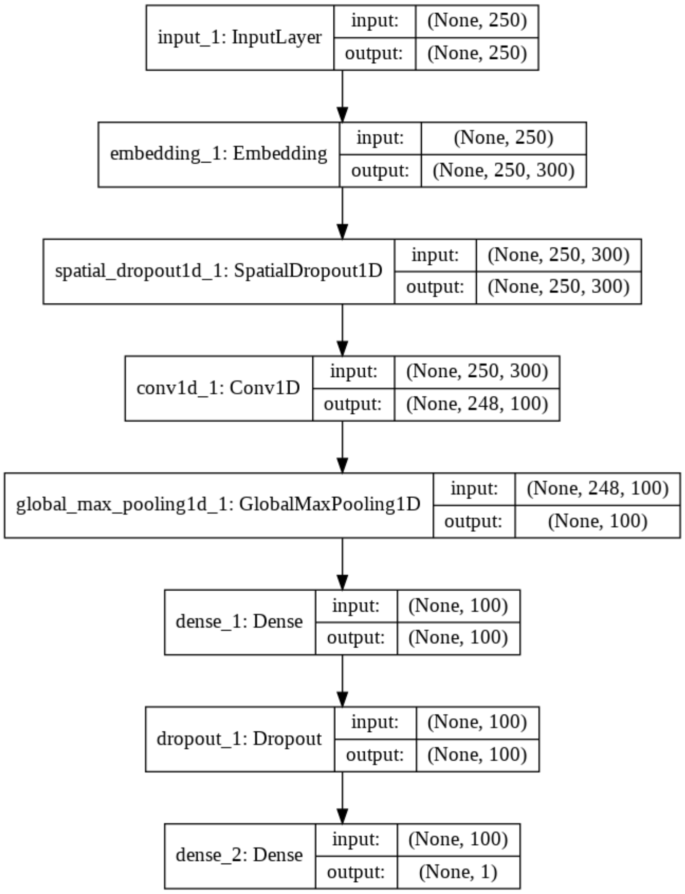

Fig. 4. CNN model baseline

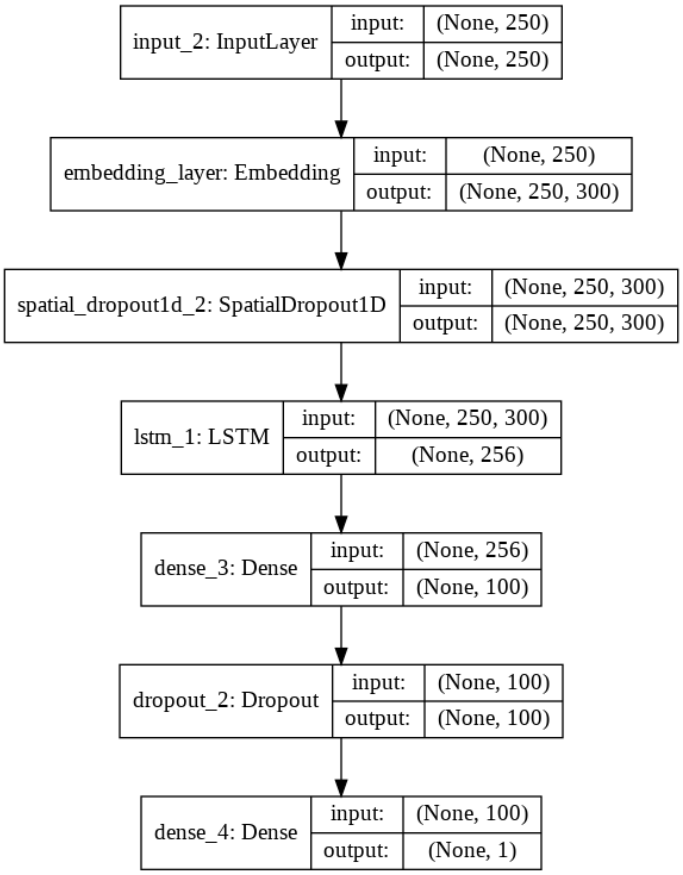

Fig. 5. LSTM model baseline

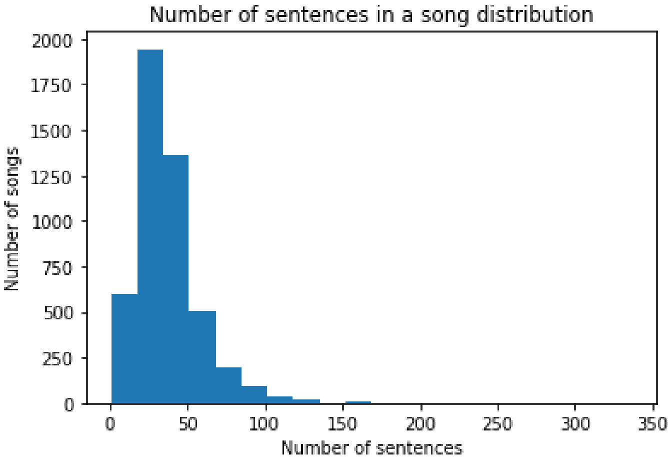

Fig. 6. Lyric's length in sentence distribution

To ensure that the loss of information will not affect the model, we choose each line has a maximum of 10 words and each song has a maximum of 60 sentences because from the distribution we can see that with these numbers, we will keep around 90% words and sentences in the dataset.

From 20,000 words from 3337 songs of the train dataset is embedded with the corresponding word vector in GloVe to build up 20,000 300-dimension vectors. Then, each word in a song is replaced by its corresponding vector. The input to the first attention layer will have shape (60,10,300) corresponding to (maximum sentences, maximum words in sentence, embedding dimensions)

Bidirectional GRUs had 64 hidden units and 128 states are output from the attention mechanism. All the hyperparameters were tuned by using the evaluation in validation set. We are working on a small dataset so to avoid overfitting we have dropout layer with rate as 0.5 and apply l2 regularizer with l2=0.05 for each layer with trainable parameter. We train the model with batch size of 512 and optimize using RMSprop with a learning rate of 0.01. To achieve the best result, we also used early stopping to stop the model if the validation accuracy did not increase for 10 successive epochs. The graph of model is described in detail with Figure 4. We carried out the classifier on GPU in Google Colab 16GB of high bandwidth memory

C. *Result:*

The test accuracies are seen in Table 1

We achieved the better result than original research but it is not too significant.

1) Attention Visualization:

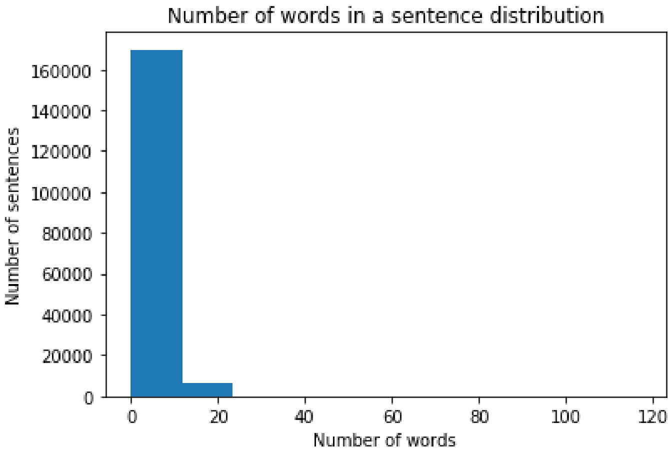

Fig. 7. Number of words in a sentence distribution

TABLE I
Our Method Initial Result

Mlpn	HAN
CNN	LSTM
75.63%	76.01%
72.4%	66.6%

V. Conclusion

To extract the semantic orientation from lyrics of a song, we have applied the hierarchical attention networks which have been applied in many documentation classification tasks. The method is applied on MoodyLyricsPN dataset, which was created by Erion Çano , containing 5,000 songs that have been classified as positive or negative. The proposed method increased the result from 75.63% to 76.01%. However, the best characteristic of the HAN model is it can visualize what it has learn in terms of the important of sentences and words for the prediction. To making the best classifier for songs emotion, in the future, we will also consider other resource such as the audio of the song and combine it with lyric-based approach.

References

[1] Christine Bauer, Marta Kholodylo, and Christine Strauss. Music recommender systems: Challenges and opportunities for non-superstar artists. 06 2017.

[2] Aziz Nasridinov and Young-Ho Park. A study on music genre recognition and classification techniques. In *MUE 2014*, 2014.

[3] Doris Baum. Emomusic - classifying music according to emotion. 01 2006.

[4] Stefan Brecheisen, Hans-Peter Kriegel, Peter Kunath, Alexey Pryakhin, and Florian Vorberger. Muscle: Music classification engine with user feedback. In *EDBT*, 2006.

[5] Andrew Demetriou, Andreas Jansson, Aparna Kumar, and Rachel M. Bittner. Vocals in music matter: the relevance of vocals in the minds of listeners. In *ISMIR*, 2018.

[6] Pasi Saari, Tuomas Eerola, and Olivier Lartillot. Generalizability and simplicity as criteria in feature selection: Application to mood classification in music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19:1802 – 1812, 09 2011.

[7] Tuomas Eerola and Jonna K. Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, 2011.

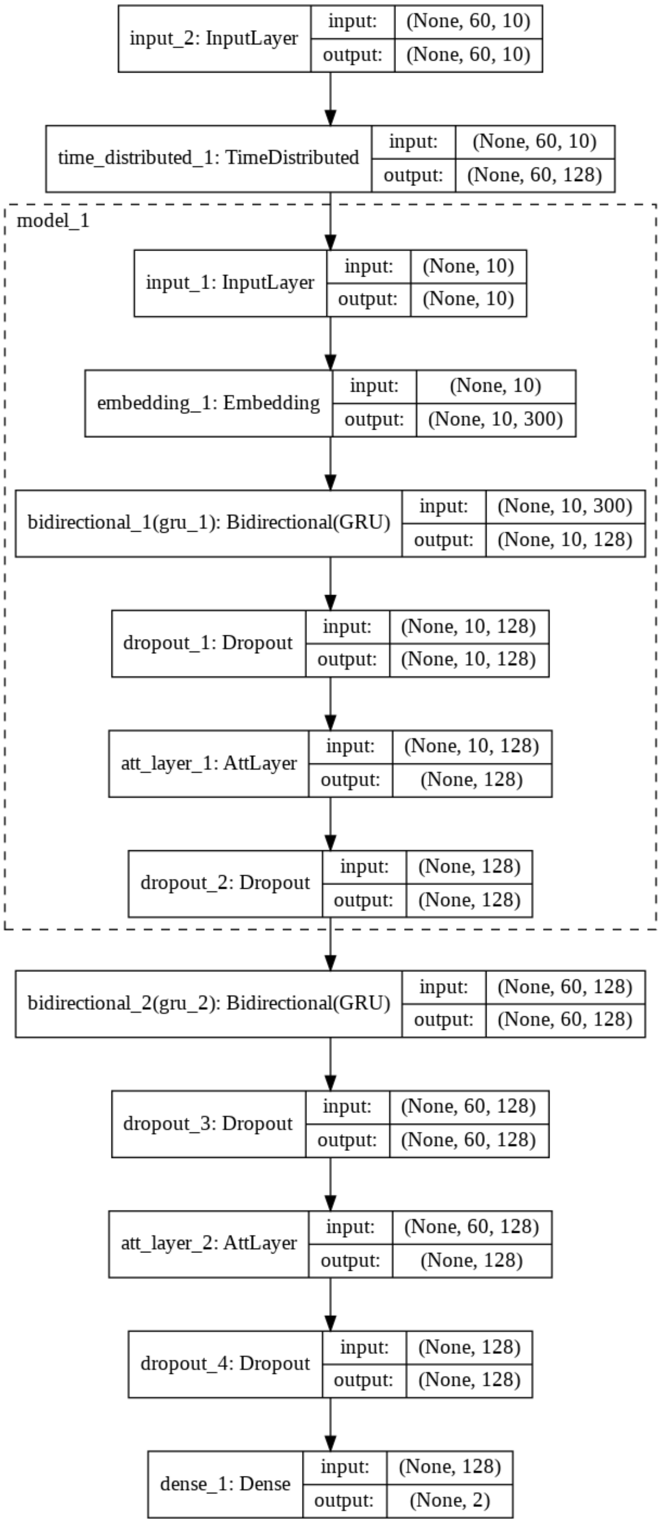

Fig. 8. Hierarchical Attention Networks model

[8] Tuomas Eerola, Olivier Lartillot, and Petri Toiviainen. Prediction of multidimensional emotional ratings in music from audio using multivariate regression models. pages 621–626, 01 2009.

[9] C Laurier, Olivier Lartillot, Tuomas Eerola, and Petri Toiviainen. Exploring relationships between audio features and emotion in music. pages

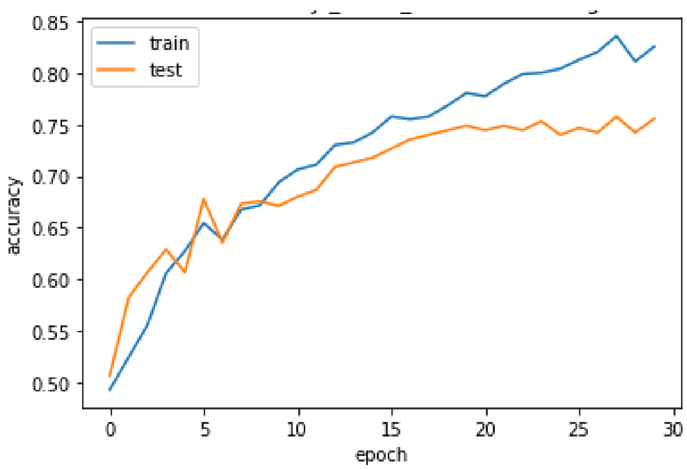

Fig. 9. Training process

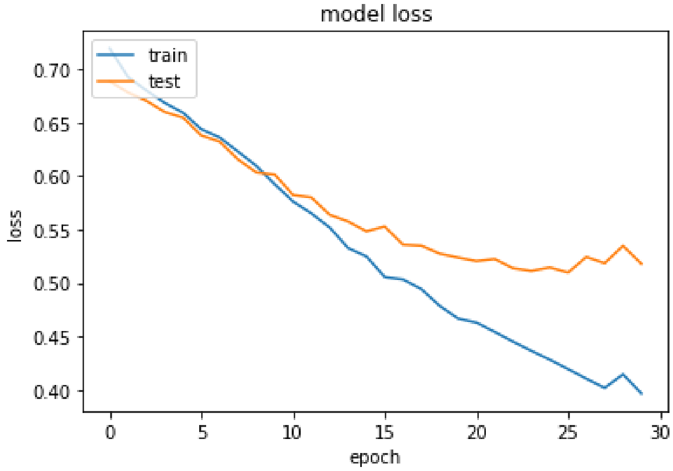

Fig. 10. Loss

260–264, 01 2009.

[10] Fika Hastarita Rachman, Riyanarto Sarno, and Chastine Fatichah. Music emotion classification based on lyrics-audio using corpus based emotion. *International Journal of Electrical and Computer Engineering (IJECE)*, 8:1720, 06 2018.

[11] Erion Çano and Maurizio Morisio. Moodylyrics: A sentiment annotated lyrics dataset. pages 118–124, 03 2017.

[12] Erion Çano. *Text-based Sentiment Analysis and Music Emotion Recognition*. PhD thesis, 06 2018.

[13] Dan Yang and Won-Sook Lee. Music emotion identification from lyrics. pages 624 – 629, 01 2010.

[14] M.D. Devika, Sunitha C, and Amal Ganesh. Sentiment analysis: A comparative study on different approaches. *Procedia Computer Science*, 87:44–49, 12 2016.

[15] Z. Hailong, G. Wenyan, and J. Bo. Machine learning and lexicon based methods for sentiment classification: A survey. In *2014 11th Web Information System and Application Conference*, pages 262–265, Sep. 2014.

[16] Neel Kant, Raul Puri, Nikolai Yakovenko, and Bryan Catanzaro. Practical text classification with large pre-trained language models. *CoRR*, abs/1812.01207, 2018.

[17] Olga Kolchyna, Thársis Souza, Philip Treleaven, and Tomaso Aste. *Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination*. 07 2015.

[18] Sebastian Raschka. Musicmood: Predicting the mood of music from song lyrics using machine learning. 11 2016.

[19] S Hardik, Dhruvi D Gosai, and Himangini Gohil. A review on a emotion detection and recognization from text using natural language processing. 04 2018.

[20] Prabu palanisamy, Vineet Yadav, and Harsha Elchuri. Serendio: Simple and practical lexicon based approach to sentiment analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 543–548. Association for Computational Linguistics, 2013.

[21] Çano, erion; morisio, maurizio. music mood dataset creation based on last.fm tags. *Fourth International Conference on Artificial Intelligence and Applications*, (AIAP 2017), Vienna, Austria, 27-28 May 2017. pp. 15-26, DOI:10.5121/csit.2017.70603.

[22] Wikipedia. https://en.wikipedia.org/wiki/FastText. Accessed: 2020-1-8.

[23] Wikipedia. https://en.wikipedia.org/wiki/Word2vec. Accessed: 2020-1-8.

[24] Medium. https://medium.com/@japneet121/word-vectorization-using-glove-76919685ee0b. Accessed: 2020-1-8.

[25] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[26] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[27] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.