Music Mood Classification

1st Tri-Nhan DO
HCM University of Science
Advanced Program in Computer Science
dtnhan@apcs.vn

2nd Hoang-Tan NGUYEN
HCM University of Science
Advanced Program in Computer Science
nhtan@apcs.vn

3rd Minh-Tri NGUYEN
HCM University of Science
Advanced Program in Computer Science
nmtri17@apcs.vn

4th Thanh-Vy HUYNH
HCM University of Science
Advanced Program in Computer Science
htvy@apcs.vn

5th Van-Tu NINH
HCM University of Science
Advanced Program in Computer Science
nvtu@apcs.vn

***Abstract*—Music mood classification is one of the most important features in music recommendation. To enhance the quality of music recommendation, the authors propose a method to classify moods of songs by combining techniques of semantic analysis in lyric. The method we utilized is adding more feature from rule based and lexicon based method to word embedding pretrain. Then the author deploy some neural network models and train in MusicMood dataset .The author manages to achieve the accuracy of 74.5% with CNN model. The result increases by 2% compared to classical machine learning method of the same dataset using Naive Bayes classification model.**

I. Introduction

According to proposed method for music classification, music can be classified according to artist, genre, instrument or mood of the song. The approach to classify music varies in different ways. One of a potential one is to use subjective human feedbacks or user tags [ref], However, a drawback of this method is that it is hard to collect dataset since users are not always willing to provide their feedbacks. Another approach is to classify based on its characteristic like audio and lyrics. However, not only does the songs characteristic but also feeling of different people based on the situations that it is played make 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 [2, 3, 4]. Although dataset collected by MIREX has high reputation and is created by many experts, the dataset is not enough for training in deep learning. MoodyLyrics, is another reliable dataset which uses valence and arousal values of the word based on Russells model. It consists of 2595 song lyrics, which is bigger than most of the current publicly available datasets. There are also others hand-labeled datasets which are created for private projects.[5]

To determine mood of a song, some studies use lexicon based, whereas some others apply machine learning approaches. Our proposed method is to combine both of the two methods, using lexicon based for feature pre-processing and using CNN, GRU, LSTM model for mood prediction. The authors experiment some pre-trained word vector extraction methods like fasttext, word2vec, GloVe to to choose the most proper feature extraction model. To validate the quality of the mood, the authors compared their models with two other researches with the same dataset [6][7]. The evaluation process reveals an accuracy of 74.5%, which increases the accuracy of Naive Bayes method by 2%. It is also higher than a Lexicon based method. Moreover. A demo website was created for users to try and expand dataset by user feedbacks.

The rest of the paper is organized as follow: Section 2 reveals some related works on music mood classification and explain why the authors choose a dataset. Section 3 introduces the authors proposed method using to analyze lyrics of a song. Section 4 presents the results and discuss some future works.

II. Related work

A. Twitter sentiment analysis

Lexicon-based method and machine learning approach are applied to analyze the variation of the public opinion about retail brands (0,) . 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 when combining the two approaches together by extracting features for Naive Bayes and SVM classifier from lexicon score

B. Music mood classification using machine learning

A recommendation system was built based on a Naive Bayes classifier (0,). 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 songs based on English keywords denoted feelings like happy or sad (0,). Textual content from social networking site is obtained and defined into structure of list of sentences, list of tokens, word form, word lemma, and associated tags. After re-defining their structure and preprocessing 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 (0,). The lexicon used in this method is created manually containing 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 the sentiment of the tweets.

E. Word Vector for Sentiment Analysis

This project proposed the approach of combining unsupervised and supervised method for capturing semantic andbetween sentiment similarities among words. The comparison 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 presented gold standard lexicon, which was produced by combining qualitative and quantitative method and under consideration of grammatical and syntactical general rule for sentiment intensity expression and emphasis, specially attuned to microblog-like contexts. The evaluation indicate 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. They are both validated by humans and is gold-standard quality.

III. Proposed Method

We apply three popular sentiment analysis methods on the same dataset.

A. Rule Based Approach

Rule based approach is used by defining various rules for getting the opinion(0,). It requires skilled experts: it takes a linguist or a knowledge engineer to manually encode each rule in NLP

The authors manually create two labelled re-defined dictionaries of words, which is sad and happy. We build two tries to store two group of mood. Each word in a trie annotated with the word s frequency and assign to 0 initially. By analyzing 1000 songs from the training labeled dataset, each song is tokenize. For each word in that song, we look up it in the trie has corresponsive mood with the song mood the word belong to. If the word is not there add it to the trie, set its frequency to 1, else if it is available, count up.

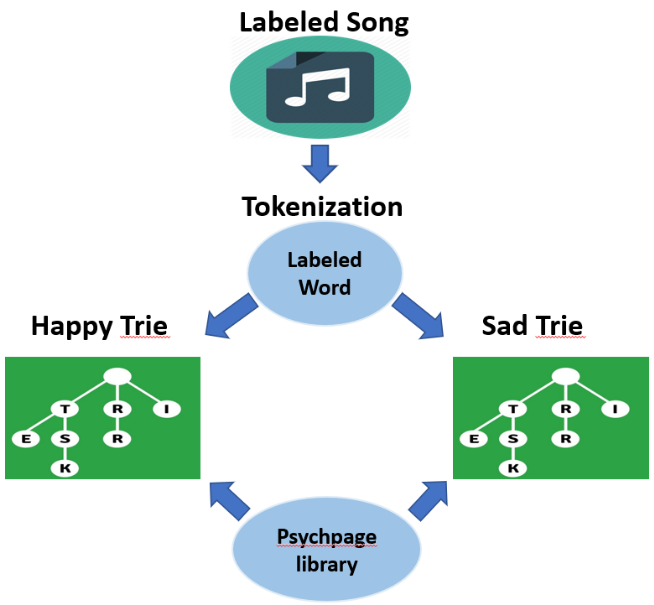


Fig. 1. Manual approach

To expand list of the dictionaries, a list of adjectives from psychpage library are compiled into the corresponding dictionary [11]. After removing stopwords, for any words that have high frequency in both sad and happy dictionary and they are nearly equivalent, we set their frequency back to 1. With a word which appear in the first trie is much higher than the other, the authors simplify their frequency ratio. Then, 200 validation data are input and sentiment conveyed by the lyrics can be obtained by aggregating the semantic orientation score. For each word in a test song, we check it in both tries to consider which trie it appear or if the word is in both trie, set its mood to the mood of trie has higher frequency. We count number of happy words and sad words in the song to find the higher mood, return that mood for the song. With this method, the efficiency and accuracy depend the defining rules and it is not flexible.

B. Lexicon based approach

In this approach, we use trained model VADER (Valence Aware Dictionary and sEntiment Reasoner) from NLTK library. VADER is a lexicon and rule-based sentiment analysis tool. It is constructed from a generalizable, valence-based, human-curated gold standard sentiment lexicon and it is fast enough to be used online with streaming data. Each of the words in the lexicon is rated as to whether it is positive or negative. We use VADER to analyses each song in the validation to checks if any of the words in the lyric are present in the lexicon. Base on the rating of those words, VADER produces four sentiment metrics represent the proportion of the song that falls into positive, negative or neutral. Because our purpose is to classify into 2 mood, each lyric of songs is divided into sentences, then we apply VADER for each sentences. The mood of a song is mood that have more sentences.

C. Machine learning approach

Although lexicon-based approach can reflect the characteristic of the unstructured text data, it depends only on the dictionary, requires powerful linguistic resources. It means that a word can only be labelled if it is available in the dictionary. The problem here is that words and adjectives extracted from the training dataset we used is not large enough to analyse words and adjectives extracted from validation data [12]. To overcome the influence of the unavailable words, the authors apply machine learning method. Machine learning approach includes training algorithm by using a training dataset and a validation dataset. This method have the advantage of not only demonstrate the high accuracy of classification but also using dictionary is not necessary.

The authors go through 3 main steps: pre-processing to standardized dataset and filter out any word does not affect the general mood of the song, then use feature engineering to transform data into flat features and finally model training on a labelled dataset.

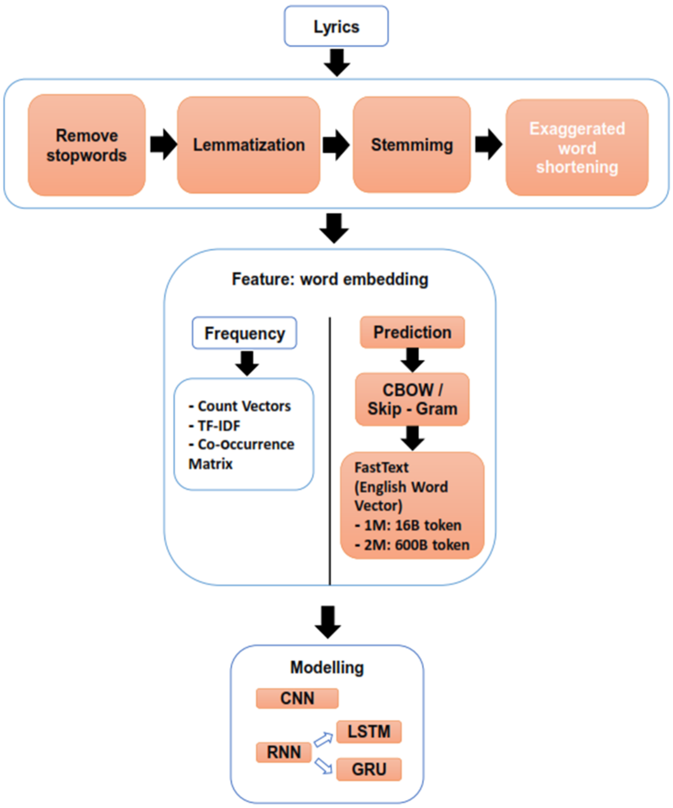


Fig. 2. Machine learning method

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

The authors apply text normalization techniques, which is lemmatization and stemming from the NLTK library to achieve the root forms of inflected words. Stems are created by removing the suffixes or prefixes used with a word. Stemming a word may result in words that are not actual words because it uses the rules to decide whether it is wise to strip a suffix. The dataset we choose is totally in English, there are two popular kind of English stemmer: PorterStammer or LancasterStammer. The authors choose LancasterStammer algorithm because it use more aggressive approach and rule. With lemmatization, a word returns an actual word of the language. It reduces the inflected words properly ensuring that the root word belongs to the language. We choose WordNet Lemmatizer that uses the WordNet Database to lookup lemmas of words.

All stopwords and punctuations is filter out, including words that exist in most songs or have very little meaning. We not only use stopword from NLTK library but also we expand the list by remove 100 words having high frequency in the dataset by utilizing the frequency distribution function in the library. To handle lengthened words like humming,the authors 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.

To expand the initial dataset, the authors add 2190 sentences divided into 2 emotions from the ISEAR project [13]. The complete dataset includes 3190 sentences. We also try to split the lyric of each song into two parts to double the dataset but this way do not improve the result much.

2) Feature engineering : In the next phase, the authors use Frequency based Embedding with two vectorization methods: Counter Vectors, TF-IDF

Counter vector of a word is vector store number of times a word has appeared in N-song, the number of unique word from N-song is the number of feature is used for training. In the dataset we choose, each song lyric have 12309 feature form from 1000 training file. 12309 unique tokens are extracted out of the corpus, they form dictionary with the size of 1000*12309. Because the number of word is quite large, the matrix is too much sparse. Python provides an efficient way of handling sparse vectors in the scipy.sparse package. All words were made lowercase by default and that the punctuation was ignored.

Common words like is, the, a etc. tend to appear quite frequently in comparison to the words which are important to a document. We down the weight of these words and make importance to words in a particular. To deal with this issue, Term Frequency Inverse Document Frequency method is applied. The TF-IDF score for each word in an song is normalized to values between 0 and 1 and is calculated by:

$$TF - IDF(word, song) = TF(word, song) * IDF(word) \tag{1}$$

- TF is the number of times word t appears in a song per number of word in that song.
- IDF is common logarithms of total of song per number of song the word appear.

If a word appear in all of the songs, the IDF of it is equal to zero, so the TF-IDF score is zero. The authors try three type of analyzer: word level, ngram level and characters level with the range of ngram from 2 to 3. It is not effective to transform the whole vocabulary. To deal with the sparseness, we set the max-feature equal to 500.

3) Model training: After extracting features, the authors apply some common classical models in machine learning like Naive Bayes Classifier, Linear Classifier (Logistic Regression), Support Vector Machine, Bagging Models, and Boosting Models.

Support Vector Machine: It is a non-probabilistic classifier in which a large amount of training set is required. Each song is plotted as a point in n-dimensional space (where n is number of features) with the value of each feature being the value of a particular coordinate. Select the hyper-plane which segregates the two mood better. SVM has a technique called the kernel trick. These are functions which takes low dimensional input space and transform it to a higher dimensional space to find out the process to separate the data based on the labels. We try with Radial basis function kernel and linear kernel . Because the number of feature extract from the dataset is over 4500, the song is linearly separable in high dimensional space, so the result with linear kernel is better. To set the fitting to training data, the authors take the gamma equal to 10. In text processing, the mood classes are often overlapping. We much improve the noise in dataset so that SVM can perform well.

Extreme Gradient Boosting: is a type of ensemble models part of tree based models. It has both linear model solver and tree learning algorithms, supports various objective functions, including regression, classification and ranking. There are three types of parameters: General Parameters, Booster Parameters and Task Parameters.

Naive Bayes Method: It is a probabilistic classifier and is mainly used when the size of the training set is less. In machine learning it is in family of sample probabilistic classifier based on Bayes theorem.

$$P(X|Y) = \frac{P(X) * P(Y|X)}{P(Y)} \tag{2}$$

So for finding the sentiment the equation is transformed into the below

$$P(S|T) = \frac{P(S) * P(T|S)}{P(T)} \tag{3}$$

with sentiment denoted by S and sentence denoted by T.

There are three types of Naive Bayes model: Gussian for features follow a normal distribution, Multinomial for discrete counts, Bernoulli for feature vectors are binary. Because we have count how often word occurs in the document by TF-IDF so the authors choose multinomial model.

4) Neutron Network Models: Featuring by Bag of word (BOW) based on frequency can't express the connections between words, it discards word order thereby ignoring the context and in turn meaning of words in the song. and Nave 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, the authors 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]. 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. The authors decided to use fastText to featuring vector because it generate word embeddings for rare words better. The pre-train set is fastText 1 million word vectors trained on Wikipedia and 2 million word vectors trained on Common Crawl. The results between two sets do not differ significantly.

The final step is to train a classifier with Deep Neural Networks. The authors run model Convolutional Neural Network (CNN), Long Short Term Model (LSTM), Gated Recurrent Unit (GRU) from keras library.

Our best result model has 4 layers: 1 embedding layers, 1 convolutional layer and 1 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 dropout 1D after the embedding layer and dropout after the dense layer but not after the convolutional layer.

From the songs lyric, we convert it to sequence of tokens and pad it to ensure equal length vectors of 70. Then we put it into the embedding layer before extract feature with convolutional layer because one-hot encoded vectors is high dimensional and sparse. Another useful property of embedding layer is it can be updated while training the neural network but we havent use it for now. We are working on a small dataset so to avoid overfitting we have used spatial dropout 1D rate as 0.3, which performs the same function as dropout however it drops entire 1D feature maps instead of individual elements. If adjacent frames within feature maps are strongly correlated then regular dropout will not regularize the activations, but this type of dropout will do. After the convolutional layer, we use global max pooling layer to feed directly feature maps into feature vectors.An advantage of that is there is no parameter to optimize in the global max pooling thus overfitting is avoided at this layer. From feature vectors, we apply fully connected and sigmoid function to calculate probability that this song will have a happy or sad 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 we use Adam optimization algorithm to optimize our parameters. We conduct the training model for 60 epochs with batch size of 64.

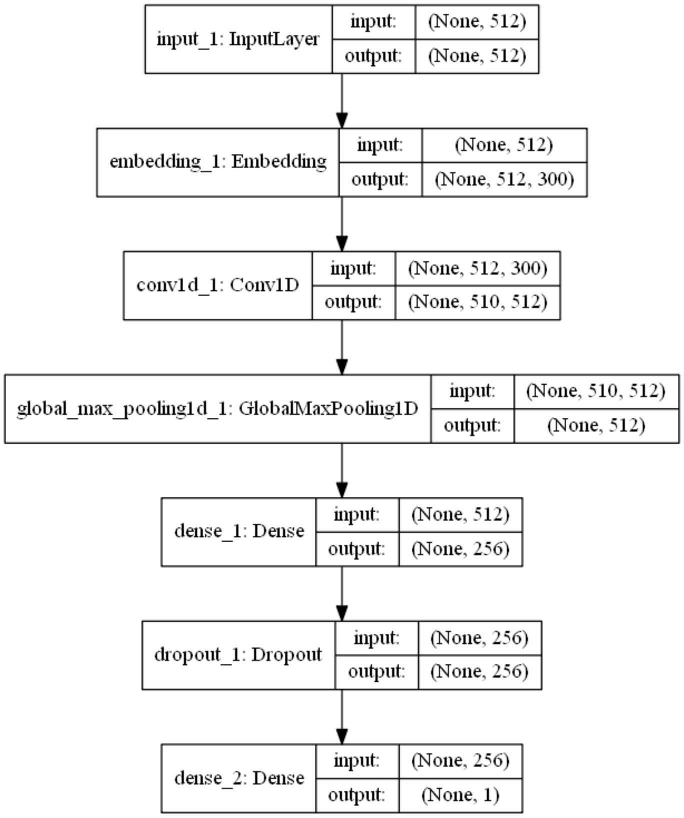


Fig. 3. Neural network model

IV. Experiment

The authors use the dataset created by Sebastion Raschka from Michigan State University. It originally contains 10000 songs downloaded from the Million Song Dataset and was filtered to remove non-lyrics and non-English songs. The remaining were subsampled randomly into a 1000-song training dataset and 200-song validation dataset. Due to the license, that dataset only contains metadata and mood label. Thus, a script was written to download it.

Configurations of computer use for this training are Intel(R) Core(TM) i5-5200U CPU 2.20Ghz, RAM 8GB

A. Rule based approach

This method yields the accuracy of 60.5%, this is not an impressive result when there are just 2 classes for classification. The accuracy when we use the training data for testing model is 92 %, we can improve by adding more rule to the model.

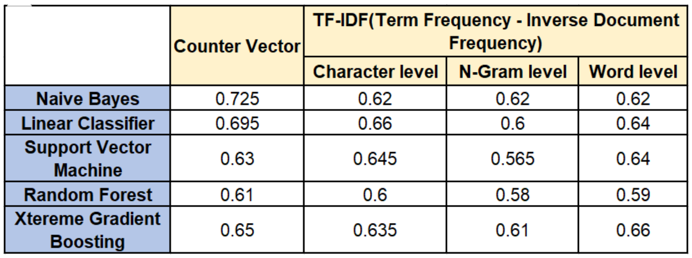
B. Lexicon based approach

By applying VADER in the validation dataset, we achieve the accuracy 64.5% without wasting time for training. Therefore, we use it to deploying a django web so that it can return the result immediately.

C. Machine learning based approach

Python library we use in this research to training model is sklearn for linear classifier, SVM, ensemble and Naive Bayes model, keras for neutron network, numpy, xgboost for Extreme Gradient Boosting. To extract feature CountVectorizer and TfidfVectorizer is use. We also use pandas for load data, nltk for pre-processing, django for create a web-app to test the result and collect data to expand dataset.

Among Non-Neutron Network model, the authors obtain the highest result with 72.5% by using Naive Bayes model.



	Counter Vector	TF-IDF(Term Frequency - Inverse Document Frequency)		
		Character level	N-Gram level	Word level
Naive Bayes	0.725	0.62	0.62	0.62
Linear Classifier	0.695	0.66	0.6	0.64
Support Vector Machine	0.63	0.645	0.565	0.64
Random Forest	0.61	0.6	0.58	0.59
Xtereme Gradient Boosting	0.65	0.635	0.61	0.66

Fig. 4. Result with Naive Bayes models

For deep learning models, the training process of each model takes about 10 minutes. We conduct the training model for 40 epochs. Use mini-batch sgd to solve the net, with batch-size of 500 because the number of train file is just 1000. Among Deep Neural Networks, CNN model yields the highest accuracy with 74.5%.

TABLE I
Our Method Initial Result

LSTM	GRU	CNN
72.5%	68%	74.5%

V. Conclusion

To extract the semantic orientation from lyrics of a song, the author applied different commonly used methods in sentiment classification. A combination of using lexicon based for feature pre-processing and Neural Network Model CNN is also proposed. The method is applied on MusicMood dataset, which was created by Sebastion Raschka from Michigan State University, containing 1000 songs for training and 200 songs for validation. The proposed method increased the result from

72.5% to 74.5% in comparison to Naive Bayes approach applied on the same dataset. However, the result shows that Neural Network Model takes longer time for training process than Naive Bayes method. Thus, the conclusion is that deep learning approach results in better performance than Naive Bayes approach if the dataset is extended. Moreover, a possible extension of this work is using feature extracted from lexicon based method for Neural Network model.

Acknowledgment

This work was supported by Ninh Van Tu for helpful comment and

References

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

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

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

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.

C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01 2015.