GRADUATE CERTIFICATE IN PRACTICAL LANGUAGE PROCESSING MODULE REPORT

Sentiment Analysis

Institute of Systems Science, National University of Singapore, Singapore 119615

ABSTRACT

Sentiment analysis is perfect for customer insights because you can identify sentiments in social mentions, hashtags, posts, comments, and videos. With sentiment analysis on community help , you can discover and attract new target audiences, and improve your brand presence

Emotion detection from text is one of the challenging problems in Natural Language Processing. The reason is the unavailability of labeled data-set and the multi-class nature of the problem. Humans have a variety of emotions and it is difficult to collect enough records for each emotion and hence the problem of class imbalance arises. Here we have a labeled data for emotion detection and the objective is to build an efficient model to detect emotion.

1. INTRODUCTION

Text-Emotion-Analysis is a project to develop rule-based and deep learning algorithms with an aim to first appropriately detect the different types of emotions contained in a collection of English sentences or a large paragraph and then accurately predict the overall emotion of the paragraph.

This business use case derives to develop a model that uses NLP techniques to accurately detect emotions in text data. The model can be used for sentiment analysis, customer feedback analysis, and social media monitoring. The model is trained on a dataset of text data that has been labeled with the corresponding emotions expressed in it.

The goal of this project is to classify the emotions in text using a Machine Learning model built with the Keras framework. The project showcases the steps involved in tokenizing text, one-hot encoding labels, defining a neural network architecture, training the model, and predicting emotions in input text.

2. DATASET

Context

After deep pre-processing of tweets in done (lemmatization, removal of stop words, etc.) This data-set is comprised of 62,015 tweets from Twitter with labelled emotions of five classes Neutral, Happy, Sad, Love, Anger.

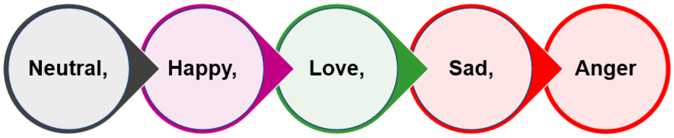


Fig. 1. Emotion Categories.

3. SYSTEM DESIGN

System Architecture

- System basically compromises of the following steps as shown in figure 1:
- Importing the data sets and other python libraries needed.
- Next we vectorize the articles in the corpus. For vectorization we use Sci-Kit Learn’sCountVectorizer to create a sparse matrix of the count of each word in an article.
- For better results we then calculate the inverse term frequency for the words using Sci-Kit Learn’sTfidfTransformers. Having got this sparse matrix we would apply classification algorithms on this vectorized word matrix to predict classes or data in test data set.
- Compare the accuracies, training times, testing times, predictions, etc. to prepare comparative report

N-Grams

In N-grams we make the given text into slices, slices of words or slices of characters. First we consider character slicing in it we can append blank character in the starting and ending of each word. Let us consider ‘_’ as a blank character. We have to select the variable N in N- grams i.e N can be one of 1, 2, 3 and so on .when we consider N = 2 it is bi-grams, when N = 3 it is tri-grams, when N=4 it is quadgrams and so on.

Example: In-character slicing for the word “HELLO” will be:- Bi-grams: _H, HE, EL, LL,O_ Tri-grams: _HE, HEL, ELL, LLO, LO_ Quad-grams: _HEL, HELL, ELL, LLO_ Generally a word or string of length L has L+1 bi-grams, tri-grams, quad-grams etc. when padded with blank characters. Example : Word slicing for the sentence “We

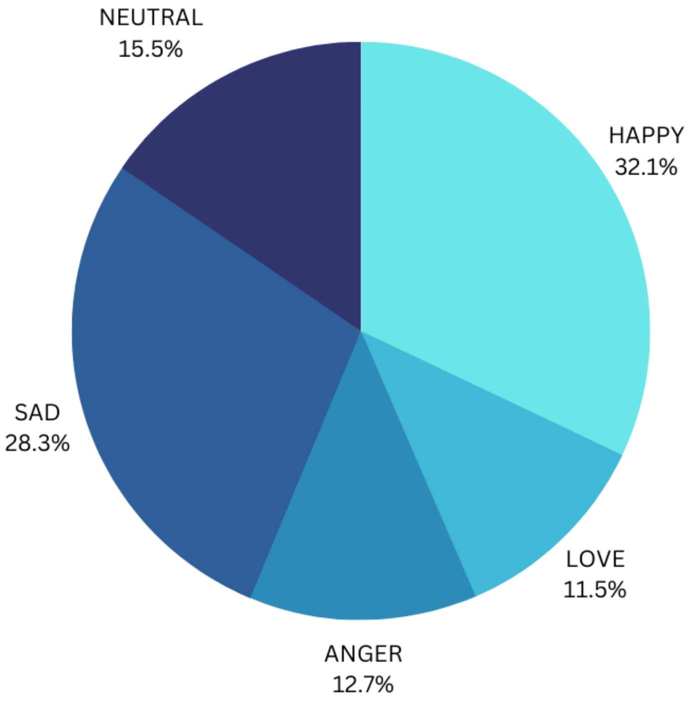
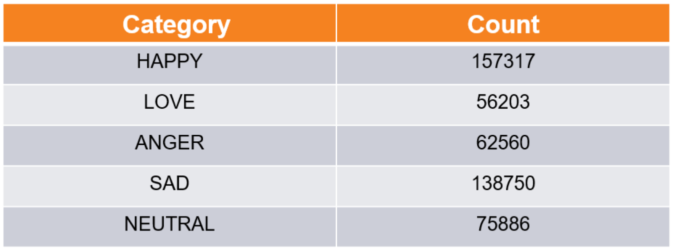


Fig. 2. Emotion Category %.



Category	Count
HAPPY	157317
LOVE	56203
ANGER	62560
SAD	138750
NEUTRAL	75886

Fig. 3. Emotion Category Counts.

have limited amount of resources to use"willbe:- Uni-grams: we, have, limited, amount, of, resources, to use Bi-grams: we have, have limited, limited amount of resources to use. Tri-grams : we have limited, have limited amount, limited amount of, amount of resources, of resources to, resources touse[12].

N-Grams usage

Normally we use some words more frequently than other. We can combine Zipf'slaw[16] with above statement and can state as : The occurrence of n most frequent word in text is proportional to 1/n. The general usage language consists on lot of words which are common. In some cases, when we are classifying same type of data then some words present in all groups. Those are not much useful while classifying the data. Generating frequency Profile:

- Data is modified by discarding numbers and punctuations, the necessary blank spaces are added to thedata.
- Then generate all possible n-grams (let's say 1 to 5) including the blanks aswell.
- Then fit the data into a hash table along with frequency such that each n-gram has its own count.
- Now sort these n-grams in descending order of occurrences then remove the count and store onlyngrams.

Usually the top n-grams are the common words we use frequently in the human language.Now comparing two texts using n-grams[13].After generating two frequency profiles one of each text,We measure the place of each n- gram in the profile with respect to another as shown in figure 2.

While classifying data into multiple groups we will take one group calculate sum of relative position of each n-grams of the text to that of the group. We will perform this on all group, then we will classify the text to a category that has the minimum sum. This is how the n-gram classification works.

4. BAG-OF-WORDS USAGE IN THE STUDY

In this model, a text is represented as the vector of its words, ignoringsentence structure and even the order of the words but keeping frequency. Frequency of each word is used as a feature to train the classifier[11].

- Allocate an integer id to every word in the text of the training set.
- For the text n, count the no. of existences of each word W and store it in X[n, m] as the value of feature m, m is the index of word W in the dictionary.

Here we have used CountVectorizer function available in the Sci-kit learn library of Python to convert the group of text documents to a sparse matrix representation [10]. There is an issue with the occurrence count that is longer the documents, higher the count values. We need to downscale the weights for words that occur in many text documents.This down scaling is called TF-IDF which stands for "Term Frequency times Inverse Document Frequency".We use the TfidfTransformer() function of the Sci-kit learn library to produce the term frequencies from the matrix of token counts. After achieving the features, we train a classifier to predict the category of an article. Here we have implemented two different classifiers, Naive Bayes and Support Vector Machines, for predicting classes of documents in test data-set

Limitations of Bag-of-Words Approach:

Bag-of-words takes into account the existences of each word, neglecting the semantics and grammar of the natural language. Thus while dealing with Natural languages, we need to take into consideration, the usage of words, semantics and meaning of the sentence the words are a part of N-Grams is one such technique, where we vectorize not one but more than one words together, which convey much more information, than just the number of occurrences.

5. SUPPORT VECTOR MACHINE

SVM is a supervised machine learning algorithm that is used to classify data as well as for the regression problems. It finds its use in most cases in the classification problems. In SVM, we represent each feature of our data-set as a point in our coordinate system. The algorithm tries to find out a hyperplane that splits the two classes with as much accuracy as possible

6. SVM USAGE IN THE STUDY

We also used Support Vector Machines(SVM), which is also a widely used classifier(although a bit slower than Naive Bayes). We create an instance of the SGD Classifier available in Sci-kit learn library and then repeat to process of training the model on training data and predicting classes for test data. To implement SVM in newspaper article classification, the first step is to remove stop words, then the punctuation marks are removed. The next step is to remove the digits since they too do not contribute to the categorization of articles. After all the data preprocessing, next step is to use bag of words or n- grams to create sparse matrix which contains the words as vectors, count as the feature. On this the support vector machine classifier is applied. The results for the same have been analyzed and discussed in the results section along with the inferences.

7. LOGISTIC REGRESSION CLASSIFIER

Multinomial logistic regression is an extension of logistic regression that adds native support for multi-class classification problems.

Logistic regression, by default, is limited to two-class classification problems. Some extensions like one-vs-rest can allow logistic regression to be used for multi-class classification problems, although they require that the classification problem first be transformed into multiple binary classification problems.

Instead, the multinomial logistic regression algorithm is an extension to the logistic regression model that involves changing the loss function to cross-entropy loss and predict probability distribution to a multinomial probability distribution to natively support multi-class classification problems.

use bag of words or n- grams to create sparse matrix which contains the words as vectors, count as the feature. On this the Logistic Regression Classifier is applied. The results for the same have been analyzed and discussed in the results section along with the inferences.

Learning Objectives

- Multinomial logistic regression is an extension of logistic regression for multi-class classification.
- How to develop and evaluate multinomial logistic regression and develop a final model for making predictions on new data.
- How to tune the penalty hyperparameter for the multinomial logistic regression model.

8. MLP CLASSIFIER

The multilayer perceptron (MLP) is a feedforward artificial neural network model that maps input data sets to a set of appropriate outputs. An MLP consists of multiple layers and each layer is fully connected to the following one. The nodes of the layers are neurons with nonlinear activation functions, except for the nodes of the input layer. Between the input and the output layer there may be one or more nonlinear hidden layers.

- **hidden_layer_sizes :** With this parameter we can specify the number of layers and the number of nodes we want to have in the Neural Network Classifier. Each element in the tuple represents the number of nodes at the ith position, where i is the index of the tuple . . Thus, the length of the tuple indicates the total number of hidden layers in the neural network.
- **max_iter:** Indicates the number of epochs.
- **activation:** The activation function for the hidden layers.
- **solver:** This parameter specifies the algorithm for weight optimization over the nodes.

9. NAIVE BAYES(MULTINOMIAL)

- The class of data set can be identified easily and quickly. It also works well for MultiClass Classification.
- When the assumption that our data is independent of the features of each other holds, then even with lesser training data Naive Bayes classifier gives excellent results.

Naïve Bayes implementation in the study We create an instance of the model available in Sci-kit learn library and then fit the training data using fit() function. Later we predict classes for test data using predict() function. If we wish to, we can calculate metrics like accuracy, rms error, etc.

10. RANDOM FOREST ALGORITHM

Random Forest is one of the most popular and commonly used algorithms by Data Scientists. Random forest is a Supervised Machine Learning Algorithm that is used widely in

Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

Random forest is a versatile machine learning algorithm developed by Leo Breiman and Adele Cutler. It leverages an ensemble of multiple decision trees to generate predictions or classifications. By combining the outputs of these trees, the random forest algorithm delivers a consolidated and more accurate result. **Learning Objectives**

- Understand the impact of different hyper parameters in random forest
- Implement Random Forest on a classification problem using sci-kit learn

11. GRU

The GRU is the newer generation of Recurrent Neural networks and is pretty similar to an LSTM. GRU's got rid of the cell state and used the hidden state to transfer information. It also only has two gates, a reset gate and update gate.

Update Gate The update gate acts similar to the forget and input gate of an LSTM. It decides what information to throw away and what new information to add.

Reset Gate The reset gate is another gate is used to decide how much past information to forget. And that's a GRU. GRU's has fewer tensor operations; therefore, they are a little speedier to train then LSTM's. There isn't a clear winner which one is better. Researchers and engineers usually try both to determine which one works better for their use case.

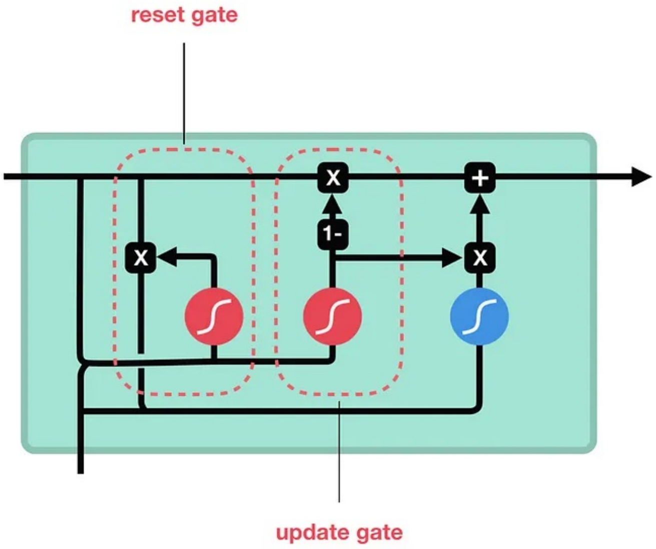


Fig. 4. GRU.

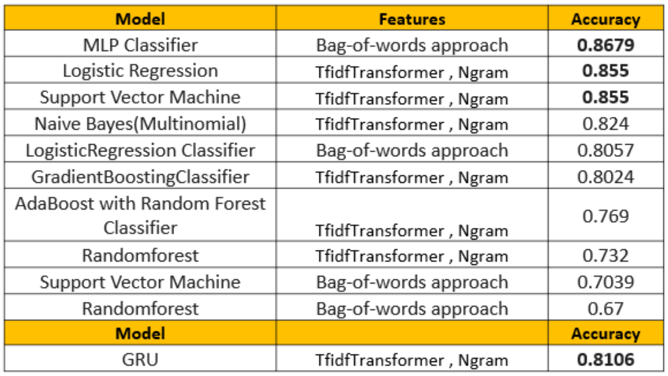
12. MODEL RESULT

Features TfidfTransformer , Ngram

- Logistic Regression is the top performer accuracy with test data 0.85 %
- Naive Bayes(Multinomial) accuracy 0.82%
- Random forest low performer accuracy with test data 0.66%
- GRU accuracy 0.81%.

Features Bag-of-words

- MLP Classifier is top performer with accuracy 0.86%
- Logistic Regression accuracy 0.80%
- Support vector machine accuracy 0.70%
- Random forest low performer accuracy with test data 0.67%



Model	Features	Accuracy
MLP Classifier	Bag-of-words approach	**0.8679**
Logistic Regression	TfidfTransformer , Ngram	**0.855**
Support Vector Machine	TfidfTransformer , Ngram	**0.855**
Naive Bayes(Multinomial)	TfidfTransformer , Ngram	0.824
LogisticRegression Classifier	Bag-of-words approach	0.8057
GradientBoostingClassifier	TfidfTransformer , Ngram	0.8024
AdaBoost with Random Forest Classifier	TfidfTransformer , Ngram	0.769
Randomforest	TfidfTransformer , Ngram	0.732
Support Vector Machine	Bag-of-words approach	0.7039
Randomforest	Bag-of-words approach	0.67
Model		**Accuracy**
GRU	TfidfTransformer , Ngram	**0.8106**

Fig. 5. Model Accuracy

13. LEARNING OBJECTIVES

- Get familiar with class imbalance through coding.
- Understand various techniques for handling imbalanced data, such as Random under-sampling, Random over-sampling, and Near Miss.
- Apply the relevant models that need to be used for each task
- Apply the major guiding principles when choosing a model for a specific task within NLP
- Decide when to and when not to use neural network based or deep learning methods for a specific task within NLP

- Analyze the time complexity involved for a specific NLP algorithm

- Pre-process textual data into suitable representation for text analytics

- Build and evaluate language models using appropriate text processing techniques for tasks like document classification, topic modeling, information extraction, etc.

- Apply deep learning techniques on large amounts of textual data to obtain high quality models

14. REFERENCES

[1] XIANG ZHANG, JUNBO ZHAO, YANN LeCUN, “Character-level Convolutional Networks for Text Classification”, Courant Institute of Mathematical Sciences, New YorkUniversity.

[2] C. dos Santos and M. Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 69–78, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics.

[3] Y. Kim. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar, October 2014. Association for ComputationalLinguistics.

[4] R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. CoRR, abs/1412.1058,2014.

[5] M. IKONOMAKIS, S. KOTSIANTIS, V. TAMPAKAS, “Text Classification Using Machine Learning Techniques “, University of Patras, GREECE, WSEAS TRANSACTIONS on COMPUTERS, Issue 8, Volume 4, August 2005, pp.966-974.

[6] Zhenya Zhang, Shuguang Zhang, Enhong Chen, Xufa Wang, Hongmei Cheng, TextCC:New Feed Forward Neural Network for Classifying Documents Instantly, Lecture Notes in Computer Science, Volume 3497, Jan 2005, Pages 232 –237.

[7] Forman, G., An Experimental Study of Feature Selection Metrics for Text Categorization. Journal of Machine Learning Research, 3 2003, pp. 1289- 1305

[8] Kim S. B., Rim H. C., Yook D. S. and Lim H. S., “Effective Methods for Improving Naive Bayes Text Classifiers”, LNAI 2417, 2002, pp.414-423

[9] BharathSriram, Dave Fuhry, EnginDemir, HakanFerhatosmanoglu, and Murat Demirbas. 2010. Short text classification in twitter to improve information filtering. In Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, pages 841–842.ACM.

[10] William B. Cavnar and John M. Trenkle, N-Gram-Based Text Categorization, Environmental Research Institute of Michigan P.O. Box 134001 Ann Arbor MI 48113-4001 pp1

[11] Zhong, S. (2005, August). Efficient online spherical k means clustering. InNeural Networks, 2005. IJCNN’05. Proceedings. 2005 IEEE International Joint Conference on (Vol. 5, pp. 3180- 3185).IEEE.

[12] Mei, J. P., Chen, L. (2014). Proximity-based partitions clustering with ranking for document categorization and analysis. Expert Systems with Applications, 41(16),7095-7105.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 3111–3119.2013.

[14] G. Lev, B. Klein, and L. Wolf. In defense of word embedding for generic text representation. InC.Bie-mann, S. Handschuh, A. Freitas, F. Meziane, and E. Mtais, editors, ”Natural Language Processing and Information Systems”, vol 9103 Lecture Notes in Computer Science, pp3550.

[15] Yiming Yang, Jan O. Pedersen, A Comparative Study on Feature Selection in Text Categorization, Carnegie Mellon University, USA, VerityInc.

[16] Zipf, George K., Human Behavior and the Principle of Least Effort, an Introduction to Human Ecology, Addison-Wesley, Reading,Mass.,1949

[17] https://www.analyticsvidhya.com

[18] https://colab.research.google.com/github/d2l-ai/d2l-en-colab/blob/master/chapter_deep-learning-computation/use-gpu.ipynbscrollTo=mwrEJrSCo-OR

[19] https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text

[20] https://www.kaggle.com/code/jarvis11/text-emotions-detection

[21] Yiming Yang, Jan O. Pedersen, A Comparative Study on Feature Selection in Text Categorization, Carnegie Mellon University, USA, VerityInc.

[22] Zipf, George K., Human Behavior and the Principle of Least Effort, an Introduction to Human Ecology, Addison-Wesley, Reading,Mass.,1949

[23] https://colab.research.google.com/github/d2l-ai/d2l-en-colab/blob/master/chapter_deep-learning-computation/use-gpu.ipynbscrollTo=mwrEJrSCo-OR

[24] https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text

[25] https://www.kaggle.com/code/jarvis11/text-emotions-detection

15. AUTHOR

Thota Siva Krishna , e0943696@u.nus.edu