Geolocation classifiers for Tweets

1 Introduction

Social networks have become an indispensable part of our lives, and the location of each user has become a critical piece of information in social networks. The task explored in this report is predicting user location on provided twitter text by using the supervised machine learning method. I will discuss three approaches and discuss the performance of them.

2 Multiview learning architecture

The original dataset published in Eisenstein et al. (2010) contains multiple twitters for a single user in the period. It has three different parts the label, a username represented by an index, and the Twitter content. Only the semantic analysis of tweets can reveal information about the user's location. At this part, feature engineering of the raw text is essential.

In the model aspect, the prediction of user geographic area can be considered as a classification problem. I will discuss three different models based on distinct feature engineering methods. One is multinomial Naive Bayes (Hutama et al., 2020) which is regarded as a model specially designed for text classification; The other two is based on Twitter User Geolocation using Deep Multiview (Do et al., 2018), first one is the MENET model covered in this report, another one is my improvement of the model to be fitted with this project.

2.1 Feature Engineering

2.1.1 TF-IDF

Term Frequency-Inverse Document Frequency (TF-IDF) (Schutze et al., 2008) is a statistical measure. A word or phrase has an enormous TF-IDF value if it has a high number of occurrences in the article and few circumstances in other articles. It determines which word is more suitable for classification. For the provided data set, the value of TF-IDF is presented as the tuple with the first place of word ID and second place of TF-IDF value. The raw data cannot use it directly since each text has a different number of tuples. The solution is to transfer to the 1-hot feature with the provided vocab file, for the word does not have value represent them as 0. This TF-IDF feature has 2038 features for each row.

I also made some improvement to this feature. There is more than one row of data about a single user. I made the combination of each row of one user, which means a user can have multiple texts for now. The reason is that from the raw data file, one user always has the same region label. A deduction on the user's dimension of data can make more valuable data to predict region. I employ the scikit-learn library to calculate TF-IDF for the combined feature but using them directly may result in stop words in the middle of the vocabulary, like "a", “and” "the", etc. They may not help us much in classification, so we can add stop words = 'English' to remove stop words.

2.1.2 Glove 300

Glove 300 is a vectorized representation of each user's tweets and makes the vectors contain as much semantic and syntactic information as possible between them (Pennington et al.,2014). The glove300 is a feature that is already provided and does not need to be processed. For each user tweet, there are 300 dimensions of a feature.

2.1.2 Node2Vec

The TF-IDF and glove300 are mainly about the text itself but not related to other users, so we employ a node2vec feature to get the information of the user network. There are two main steps of generating the node2vec component, use the networkx library to create the user network and then use the node2vec library to produce the vector with the provided user network.

In building a user graph, I use a method similar to Pennington et al.(2014): The first part is the direct relationship. If one user "@" another use and both of them are in the dataset, then add one edge between them. The weight of the edge equals the number of mentioned times between them. The second part is the indirect relationship. If one user "@" one user may or may not in the dataset, however, another user also said this third user, then add one edge between the two users and the weight equals to the time they mentioned about the third user. There is a critical part in the user graph building, considered the "celebrity" (have a vast number of connections). For the indirect

relationship, if there is one celebrity among the mentioned users, there will be many connections of people, which is meaningless. The thresh hold number for choosing the celebrity is essential. I set the thresh hold to 50. If one user has more than 50 users talked about, remove all connections with the relation users. It removes at least 14 meaningless relationships. The visual graph shows in figure1.

Figure 1 – Thresh hold = 50, 3126 users in the graph.

When I put the thresh hold to zero, which means for the indirect relationship, if the third user is not in the dataset, there will not be an edge between the two users. It removes at least 1095 relationships, contains a lot of helpful information. The visual graph shows in figure2.

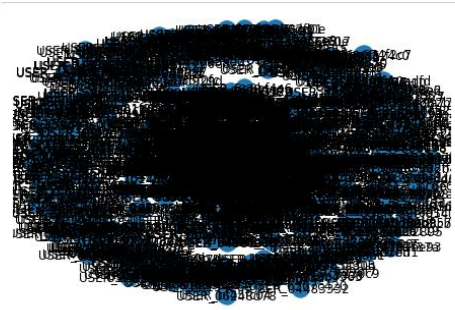

Figure 2 – Thresh hold = 0, 1045 users in the graph.

Comparing to the fig1, the side nodes are removed. We cannot get information about the indirect relationship, and the resulting graph data will become featureless.

2.2 Model Architecture

2.2.1 Multinomial Naïve Bayes

I imply a multinomial naive Bayes model for this question since I learnt from Hutama et al.(2020) that naive Bayes has a better performance than decision tree in text classification problem. After adjusting the parameters, I chose to use this model directly without any parameters to achieve the best performance. I train the model on two datasets. The first one is the origin TF-IDF feature at the training data aspect, which has 2038 dimensions, and the other one is the TF-IDF feature generated by the sklearn library, which has 300 dimensions. The origin TF-IDF feature performs better since the 300 dimensions are not enough for gathering useful information of all the content.

2.2.2 Multi-entry neural network (MENET)

Many feature sets can use for classification in this project. Still, a straightforward model implemented in the sklearn library such as Naive Bayes, neural network or decision tree just accepted one training dataset and cannot use all the information we get. There is one method mentioned by Do et al. (2018), use the multi-entry neural network can make use of all of the features. The general framework of the model shows in figure3.

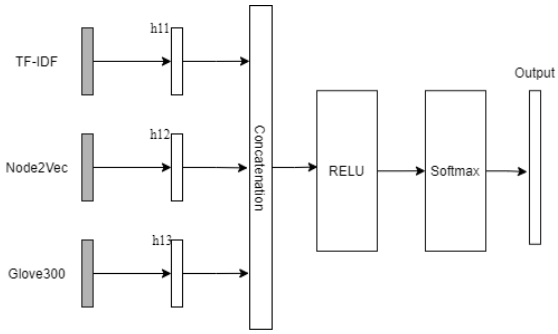

Figure 3 – Architecture of MENET

The model receives three different inputs, the 2038 dimensions TF-IDF feature, the node2vec feature and the glove300 feature covered in the feature engineering part. Each input needs to go through the hidden layer with pre-defined hidden units (shown in table I) with no activation function. For this part, Do et al. (2018) defined it as one branch.

	Number of hidden units
h11	150
h12	150
h13	30

Table I- Hyper parameter setting for hidden units of branch h11, h12, h13.

Each branch can be regarded as a dimensionality reduction function to avoid overfitting. After passing through their own defined hidden layers, the model combines the layers' output by concatenating them before entering the network. Then the model implies one fully connected hidden layer with Rectified Linear unit (RELU) activation function, trying to convert original feature vectors to a uniform

feature space. Finally, the output layer implies the SoftMax to convert scores into class probabilities and set the weight decay regularizer with λ equals to 0.1 to avoid overfitting. At the stage of compiling the model, using Stochastic Gradient Descent (SGD) algorithm as the optimization function and the loss function is categorical cross-entropy since it is considered a classification problem. Monitoring the loss value and use the early stopping method to avoid overfitting when training the model.

This MENET model is hard to implement by the sklearn library. Keras is an ideal library for implementing MENET as it is the modular and easily scalable method (Team, 2021). It allows defining different input feature and activation functions for different layers.

2.2.3 Improved MENET based on data feature

There are two types of TF-IDF feature covered in the feature engineering section. One is content based used in MENET. The other one is user-based and will be used in this part. The improvement of the MENET model is mainly about changing the input training feature but not on the framework of the model. The TF-IDF and glove 300 feature has some common characteristic; both are about the information of the content. The glove 300 feature is removed from the origin MENET model; moreover, change the TF-IDF and node2vec feature to the user base. At this part, each user has one feature instead of each content has one feature. There are 3400 users in the training set, which means there are 3400 pieces of training data and labels. At the training stage, using the ADAM optimization algorithm with an initial learning rate of 0.0001 and anneal the learning rate as the training proceeds.

3 Results and Analysis

The accuracy of the three different models shows in table II. The accuracies of these three models are all above the 0-R baseline.

Dataset / **Model name**	Training	Test
Naïve Bayes	49.1%	46.3%
MENET	43.3%	39.1%
Improved MENET	79.1%	51.2%

Table II- Classification model prediction accuracy based on training and test dataset.

As the data in the table, the improved MENET has the best performance with 50% accuracy on the development data set. We can see that the same user will only have one identical label from the source file regardless of how many tweets he has sent. Therefore, it is more efficient to make the classification by users than by tweets. In particular, for the node2vec feature, each node is a user, and if each tweet is a unit of data, then different tweets sent by a user will have the same data. In the training dataset, there are 133,796 tweets in total but only 3,400 users, which means that there are 130,396 duplicates. It would significantly impact the original MENET model as there would be a lot of unproductive data. As a result, the MENET model does not perform well with a 39% accuracy. The multinomial Naive Bayes has 46.1% on the development dataset. The reason might be that it can only consider the text itself but not explore the relationship between users. In other words, the Naive Bayes model does not make use of the data ultimately.

4 Ethical Issues and Improvement

Using machine learning to predict location based on tweets is a reasonable proposition. But if people with ulterior motives use a perfect prediction model, it could have adverse social consequences. Because the data needed for this model is so readily available, they can quickly get users' location and use it for illegal marketing, illegal promotion, etc., which is harmful to users and unethical.

To reduce the occurrence of this kind of behavior, I think it is crucial to protect users' personal information. Even if someone else gets the tweets, they only get tweets content but no other information. It means that the user's data cannot be obtained through the user ID in our project.

5 Conclusion

Targeting Twitter users has always been a difficult task because of the cacophony of text content and the range of coverage. Although there is a lot of literature on efficient methods, most of them only work on their own dataset and the accuracy fluctuates if the data changes. No method has yet been established that can be used with high accuracy irrespective of the dataset. In this paper, the enhanced version of MENET has the best performance on the dataset provided. It inferred the location of the

user based on word frequency and information about the user's network and achieved the best performance in three different models.

References

Eisenstein, J., O'Connor, B., Smith, N. A. and Xing, E. P. (2010) A latent variable model for geographic lexical variation. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010), pages 1277–1287. Cambridge, USA.

Hutama, N., Lhaksmana, K., & Kurniawan, I. (2020). Text Analysis of Applicants for Personality Classification Using Multinomial Naïve Bayes and Decision Tree. *JURNAL INFOTEL*, *12*(3), 72-81. https://doi.org/10.20895/infotel.v12i3.505

Do, T., Nguyen, D., Tsiligianni, E., & Cornelis, B. (2018). Twitter User Geolocation using Deep Multiview Learning. Retrieved 9 May 2021, from.

Manning, C., Raghavan, P., & Schütze, H. (2008). An Introduction to Information Retrieval. Cambridge University Press.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543.

. Rahimi, T. Cohn, and T. Baldwin, "A neural model for user geolocation and lexical dialectology," in Annual Meeting of the Association for Computational Linguistics, 2017, pp. 209–216.

Team, K. (2021). Keras: the Python deep learning API. Keras.io. Retrieved 10 May 2021, from https://keras.io/.