**1. Introduction**

Nowadays, social media such as Twitter, becomes one of the most popular approaches for people to share and acquire information. There are tons of messages being posted online with potential hidden information related to the user's location, events around and so on. Automatic geolocation is one of the common analytic methods that analyzes possible posted location of online texts, and is particularly useful in fields like disaster detecting [1] and stock market monitoring [2].

In this paper, we compare the impact of different feature engineering strategies and classifiers on the effectiveness of text-based tweets geolocation prediction.

**2 Related Work**

Text Approach uses text information and several different approaches to predict location. Some of them combining traditional machine learning classifiers with different feature extraction strategies, for instance, extracting “local” entities [4] from text using clustering algorithms like DBSCAN, selecting words from probabilities prospective [5], or using word embedding[6, 7, 8, 9]. Other common approaches includes multilayer model [3] and Neural Network based method [10, 11].

**3 Text Data**

The text data is directly collected from Twitter using the Twitter API, used only for research purpose that follows the Twitter Terms of Service. We choose tweets from 4 cities in Australia (Melbourne, Sydney, Perth and Brisbane) with same frequency. The text dataset consists of 248,828 tweets and is being randomly divided into train (103,364), development (37,316) and test (108,148) set, where only train and development sets have correct location label that could be used for evaluation.

**4 Experiment Approach**

**4.1 Data Preprocess**

The main purpose of preprocess is to filter out infrequent/irrelevant words in text, such as url, emojis, @usernames, numbers, punctuation symbols and stop words, short words are preserved since Roller [12] found them could contribute on geolocation prediction. Also, all left words are transformed to lowercase stem form using Snowball Stemmer to eliminate variations of same word.

After preprocessing, the size of unique words in train, development, test dataset is reduced to 58,950, 33,524 and 60,619 respectively.

**4.2 Feature Engineering**

Although preprocess help eliminate part of words, the size of unique words left is still so large that is time consuming to build model and contains lots of noises, such as words only appear once or words that equally distributed around 4 cities. Therefore, we performed feature engineering to help select a small subset of potential informative words.

Also, the course COMP30027 provides us the top (10, 50, 100) related words based on the rank of Mutual Information and Chi-Square score.

**4.3 Classifier**

According to previous work, we choose majority-class and stratified Dummy classifier, linear Support Vector Machine [5] multinomial Naïve Bayes [3, 5, 8] and Logistic Regression classifier [5] as our classifiers.

**5. Experiment Results**

**5.1 Experiment Settings**

We choose the baseline as the results from Dummy classifiers and other classifier using Mutual Information features provided from COMP30027, and since test data does not include real class label, we only evaluate our experiment with development data.

**5.2 Evaluation Metrics**

Here we use Accuracy (ACC) and Time defined below to measure the effectiveness of different feature engineering strategies and classifiers.

ACC = # of correctly identified instances / # of total instances

Time = time to train model + time to predict model

**5.3 Results**

Table 5.1 shows the effectiveness of baseline methods, the highest ACC is represented with bold font. Among baseline methods, MNB on highest 100 words of MI achieves the best accuracy 30.78% with a considerable short time 1.14 second.

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Feature Engineering | ACC | Time |
| Dummy(majority) | - | 0.25 | 0.1266167163848877 |
| Dummy(stratified) | - | 0.25176867831493194 | 0.14670515060424805 |
| MNB | MI (top10) | 0.29491370993675636 | 0.6087250709533691 |
| MI (top50) | 0.30134526744559975 | 0.795403003692627 |
| MI (top100) | **0.3078304212670168** | 1.1424529552459717 |
| LinearSVM | MI (top10) | 0.2948869117804695 | 75.24470615386963 |
| MI (top50) | 0.2979151034408833 | 74.60034322738647 |
| MI (top100) | 0.3038642941365634 | 134.71496891975403 |
| Logistic Regression | MI (top10) | 0.29491370993675636 | 47.14589810371399 |
| MI (top50) | 0.30062171722585485 | 180.77869415283203 |
| MI (top100) | 0.30405188123057136 | 384.2845768928528 |

Table 5.1 evaluation matric for baseline method

**5.4 Interpretation**

**5.5 Error Analysis**

**6. Conclusions**

Reference:

[1]: Ashktorab, Zahra, Christopher Brown, Manojit Nandi, and Aron Culotta. "Tweedr: Mining twitter to inform disaster response." In *ISCRAM*. 2014.

[2]: Mittal, Anshul, and Arpit Goel. "Stock prediction using twitter sentiment analysis." *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)* 15 (2012).

[3]: Tang, Haina, Xiangpeng Zhao, and Yongmao Ren. "A multilayer recognition model for twitter user geolocation." *Wireless Networks* (2019): 1-6.

[4]: Cheng, Zhiyuan, James Caverlee, and Kyumin Lee. "You are where you tweet: a content-based approach to geo-locating twitter users." In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 759-768. ACM, 2010.

[5]: Pappas, Konstantinos, Mahmoud Azab, and Rada Mihalcea. "A Comparative Analysis of Content-based Geolocation in Blogs and Tweets." *arXiv preprint arXiv:1811.07497* (2018).

[6]: Lim, Kwan Hui, Shanika Karunasekera, Aaron Harwood, and Yasmeen George. "Geotagging tweets to landmarks using convolutional neural networks with text and posting time." In *Proceedings of the 24th International Conference on Intelligent User Interfaces Companion (IUI’19)*. 2019.

[7]: Ajao, Oluwaseun, Deepayan Bhowmik, and Shahrzad Zargari. "Content-aware tweet location inference using quadtree spatial partitioning and jaccard-cosine word embedding." In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 1116-1123. IEEE, 2018.

[8]: Do, Tien Huu, Duc Minh Nguyen, Evaggelia Tsiligianni, Bruno Cornelis, and Nikos Deligiannis. "Twitter user geolocation using deep multiview learning." In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6304-6308. IEEE, 2018.

[9]: Thomas, Philippe, and Leonhard Hennig. "Twitter geolocation prediction using neural networks." In *International Conference of the German Society for Computational Linguistics and Language Technology*, pp. 248-255. Springer, Cham, 2017.

[10]: Rahimi, Afshin, Trevor Cohn, and Timothy Baldwin. "A neural model for user geolocation and lexical dialectology." *arXiv preprint arXiv:1704.04008* (2017).

[11]: Rahimi, Afshin, Trevor Cohn, and Tim Baldwin. "Semi-supervised user geolocation via graph convolutional networks." *arXiv preprint arXiv:1804.08049* (2018).

[12]: Roller, Stephen, Michael Speriosu, Sarat Rallapalli, Benjamin Wing, and Jason Baldridge. "Supervised text-based geolocation using language models on an adaptive grid." In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1500-1510. Association for Computational Linguistics, 2012.