COMP30027 Project 2 Report

Simpler is better: a preprocessing technique for short text geotagging

Anonymous

1. Introduction

Geotagging refers to the problem of identifying the location from which textual data originates from. Given the proliferation of portable smart devices and the popularity of social media in the 21st century, work in this area has important applications in many fields. While typical approaches to geotagging involve classifying textual data from around the globe, this paper will focus on a simplified version of the problem, where the aim is to predict the geographic origin of tweets sent from one of four Australian cities – Melbourne, Sydney, Brisbane or Perth.

1.1. Previous work

Most studies on geotagging attempt to approach the problem at a global level, and hence tend to rely on geodesic grids that map segments of textual data to specific parts of the world (Wing and Baldridge, 2011). Using grids of fixed or variable sizes, these techniques can typically yield to an accuracy rate of between 50-60% when paired with strong feature selection and models ranging from naïve bayes to advanced natural language processing techniques (Brunsting, Sterck, Dolman and Sprundel, 2016). Pappas, Azab and Mihalcea (2018) compared various feature selection methods such as Information Gain Ratio (IGR), Word Locality Heuristic (WLH) and the construction of local lexicons for each location class and found that selecting the top 30% of features based on WLH scores lead to the highest classification accuracy.

2. Methods

This section will outline the techniques that we used for our geotagging solution. Section 2.1 will provide a conceptual overview of our solution's methodology, section 2.2 will outline our feature weighting techniques, and section 2.3 will highlight our data pre-processing strategy.

2.1. Overview

Our approach relies on heavy pre-processing, a simple count vectorizer and a multinomial naïve bayes implementation to classify the origin location of a given tweet. The solution has three core steps - first, the raw textual data is pre-processed and enhanced to provide a more informative feature set, then it is vectorized using a count vectorizer that encodes the string data into a sparse feature matrix, and finally this encoded data is inputted into our naïve bayes classifier, which outputs the final class predictions. Our classifier utilises Laplace smoothing with a parameter of 1 and takes the form of multinomial naïve bayes, which is better suited to discrete feature sets involving feature counts.

2.2. Feature weighting

To simplify the problem, our solution treats each individual term present within a tweet as a potential feature. Though this methodology is simpler and potentially less informative than other feature extraction techniques such as ngrams or word-pairs, it runs faster and still stands to provide important information about terms that are geographically indicative.

In our solution, we use two techniques to identify terms that are potentially informative about location, with the first being the Word Locality Heuristic (WLH), and the second being a simple heuristic regarding twitter data. To calculate the WLH score for a given term, we first find the conditional probability of it occurring in a given city, and then divide this value by the probability that it occurs in any city. The final WLH score for a term is the maximum such value across all cities. This score thus gives a good indication of how 'exclusive' a given term is, and therefore how well it can serve as a predictor of location – terms with WLH scores around 1 occur evenly across all cities, while terms with higher WLH scores occur in certain cities more than others. Our second heuristic, the 'twitter heuristic', is based on the notion that usernames and location tags are naturally informative about the geographic origin of a tweet. Thus, these features are also considered as more important.

2.3. Preprocessing

Instead of implementing feature selection during pre-processing, where the dataset is cut down to only the more informative features, our pre-processing stage keeps the raw data and enhances it by taking into account the two weight heuristics discussed in section 2.1. For the twitter heuristic we identify usernames and location tags within individual tweets using regex, and then alter the tweet by repeating the identified features a set number of times. This serves the purpose of 'emphasising' usernames and location tags, which is analogous to assigning them a higher feature weight. For the WLH metric, we identify all terms with a WLH score above a certain threshold, and then search each individual tweet for embedded high WLH terms. We then alter the tweet again by repeating the identified terms a set number of times. This strengthens the indicative power of each tweet by highlighting the individual terms that are the most indicative. This process is repeated for both the train and dev sets.

3. Results

Method	Accuracy
Baseline NB	34.23%
Tfidf with NB	34.31%
Tfidf with LinearSVC	34.08%
Tfidf with logistic regression	33.54%
Tfidf with decision tree	31.26%
Preprocessing with NB	**36.25%**
Preprocessing with LinearSVC	33.59%
Preprocessing with logistic regression	35.34%
Preprocessing with decision tree	31.90%

Table 1: Model performance using various classification and feature weighting methods

Table 1 outlines the performance of our various classifiers over the development dataset. The best performance was achieved using our pre-processing methodology paired with a simple count vectorizer and multinomial naïve bayes. Surprisingly, this methodology lead to stronger results than more complex classifiers and feature selection techniques. Some other techniques that were attempted but not included within this results section because of low performance include ensemble learning methods such as stacking, voting and boosting – these tended to perform only worse than any individual classification model.

4. Discussion

One of the surprising aspects of our results was that naïve bayes turned out to be stronger than any other classifier. This can be attributed to the feature selection and encoding techniques that we relied on for our solution, as both count vectorizer and tfidf disregard more complex features present within the text data such as ordered word dependencies. The classification task thus becomes that of identifying the highest number of individually indicative features with the highest accuracy. Given the independent representation of these features, naïve bayes excels while more complex classifiers such as linear SVCs and logistic regression are hampered by their tendency to overfit.

While typical approaches to geotagging use some form of feature selection to narrow down the feature set, we opted to keep all of the raw data and to enhance it instead. Though it is true that many of the features with low WLH scores could have been disregarded with no negative consequences, we found that filtering terms based on WLH scores across the entire dataset tended to only hamper performance. The reason for this is because, as previously mentioned, our approach to geotagging boils down to a simple feature detection task, and so filtering terms runs the risk of losing features that could otherwise still be valuable when classifying the test set. By enhancing the raw data with a larger count of geographically indicative features, we manage to not only weight the feature set in favour of these indicative features, but to also retain as much information within it as possible.

An error analysis of our implementation reveals that misclassified tweets are mostly ones that contain almost no geographically indicative terms, and that our model's performance is therefore approaching the limits of what is possible for geotagging raw textual data. Nevertheless, there are still certain errors that can be avoided with a more complex model. For one, our classifier fails to correctly identify features that occur in slightly different forms in the train and development sets – for example, the occurrence of the term 'swanbournebeach' in the train set does not aid the classification of a tweet with the term 'swanbourne beach' in the development set. In addition, because of its reliance on unigram features, our model is more likely to fail at classifying tweets that contain a direct reference to a location with a name that has more than one word – for example 'Surfers Paradise, Gold Coast'. Both of these issues could be solved in the future through the use of more advanced natural language processing techniques that are capable of accurately identifying terminology beyond the level of simple unigram string matching.

5. Conclusions

Our work demonstrates that naïve bayes is a strong classifier for the task of geotagging, and reaffirms the notion that sometimes simpler methods are better methods. In our experimentation, we found that stronger performance came not from improving the classifier itself, but rather from specific actions that could be performed during pre-processing to improve the feature set. Our final results are competitive, and approach the boundary of what is possible using a unigram feature set for geotagging.

References

Brunsting, S., Sterck, H.D., Dolman, R., & Sprundel, T.V. (2016). GeoTextTagger: High-Precision Location Tagging of Textual Documents using a Natural Language Processing Approach. *CoRR, abs/1601.05893.*

Pappas, K., Azab, M., & Mihalcea, R. (2018). A Comparative Analysis of Content-based Geolocation in Blogs and Tweets.

Wing, B., & Baldridge, J. (2011). Simple Supervised Document Geolocation with Geodesic Grids. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL): Human Language Technologies (HLT)*. p. 955–964.