

Language detection in tweets

Siddharth Subramanian

University of Texas at Austin

`sid@cs.utexas.edu`

Abstract

This paper describes an approach to tackle the problem of language detection in tweets. Analysis of Twitter data can be difficult because of the constraint on the length of tweets, and also because of multi-lingual posts. Knowledge about the language of a tweet can help greatly in processing the data better – especially in tasks like Twitter sentiment analysis. An important task in training machine learning algorithms for this task is to get labeled data. This paper describes an approach to automatically label twitter data in an unsupervised manner. The labelled data obtained by this unsupervised labeling algorithm is then used to create a language model, which is utilized to detect languages in tweets. The technique presented in this paper gives promising results, and can be easily extended to other languages with minimum effort. Further extensions to the labeling approach using geo-located tweets are suggested; training our model using tweets collected from “reliable” regions that have a high proportion of speakers of a language further increases the accuracy of language detection.

1 Introduction

Twitter is a microblogging site which allows users to post short messages (which are no more than 140 characters long). Since this service is very popular, performing sentiment analysis over twitter posts is a cost-effective way of understanding the opinion of general public about a particular issue or product. Sentiment analysis involves a lot of text processing

and training data. However, one of the main reasons for low accuracy in sentiment analysis is that the posts are multi-lingual, and the system is trained usually using only one language. This is where language detection in twitter posts becomes important.

Language identification in text is not a new problem, though. The usual character n-gram approach trained on huge amounts of text gives accurate results in most cases. This approach does not work well on twitter data because of two main reasons: (1) the posts are very small and we do not have enough character n-grams from the post to characterize the language, (2) because of constraints on length of tweets, users create new words (for example, *smthng* for “something”) which results in an entirely new style when compared to the formal text used for training data (like EuroParl).

Obtaining labeled data from twitter is a laborious task. Even though there is a *lang* field in Twitter API’s ¹ metadata, it is often inaccurate and is not a reliable method for identifying the language of twitter posts. One way to obtain large amounts of labeled data is to find the proportion of valid word types of a particular language L in the tweet, and to assign that label L if the proportion is significantly high. This forms the basis of unsupervised labeling algorithm, which is explained in subsequent sections.

2 Related work

A lot of work has been done in analyzing the language of tweets. Hong et al. (2011) conducted a few

¹<https://dev.twitter.com/docs/api/1/get/search>

experiments on a very large number of tweets (62 million), which were aimed at characterizing Twitter data. It was found that only around 50% of the tweets are in English, showing that training models only in English would not be effective for analyzing Twitter data. A study conducted by Semiocast (2010) reveals that English, Japanese, Portuguese, Malay and Spanish are the top 5 most used languages in Twitter, and that English constitutes only half of the tweets from around the world.

In Han and Baldwin (2011), a method to identify and normalize out-of-vocabulary words is described. An ill-formed word is first detected using dependency features of the Stanford parser (Klein and Manning, 2003; de Marneffe et al., 2006). A set of suggestions are generated, and features like lexical edit distance, phonemic edit distance, and longest common subsequence are used to select the most appropriate candidate. However, the study is limited to the English language only. The work by Tromp and Pechenizkiy (2011) studies the problem of language identification on short text like tweets. A graph-based approach which considers character trigrams as well as the order of words is suggested to identify the language.

Ceylan and Kim (2009) deals with the problem of language identification of search engine queries, which is similar to our research because both these problems deal with small text inputs. It is pointed out in their work that the word n -gram approach, with $n > 1$ would not help in language identification of search queries. A decision tree classifier is built by using 3 features: character n -gram, word-based approach, and morphological feature (prefix and suffix), for detecting the language of the query. It is also shown that when the classifier included geographical information of the users, the performance received a boost. This could be useful information in our problem as well, because there are a lot of geo-tagged tweets which contain information about the location of users, and would be interesting to see the impact on the accuracy of language detection in twitter domain.

Carter et al (2012) used labeled data and other twitter parameters for language detection. But this method relies on a lot of factors other than the text in tweets, and may be time-consuming. One of the contributions of this paper is that, a labeled dataset

used for this research is published on the web, which is useful for further research on Twitter analysis. This work improves the language detection by considering other factors like blogger's previous tweets (blogger prior), the conversation in which the tweet occurs (converstation prior), tags used in the tweets (tag prior) and URLs mentioned in the tweets (link prior). Even before analyzing the actual text in the unlabeled tweet, a prior probability distribution is estimated by using these factors. This is then combined with the language detected for the actual tweet to give the final prediction. Some important results of this research work are: (1) accuracy of language detection increases when the system is trained on tweets rather than formal text, (2) exploiting the priors mentioned above gives significant improvement in the accuracy of language prediction of a tweet.

The research by Chen (2011)² employs unsupervised method for filtering non-English tweets. In this method, Expectation-Maximization algorithm is run on tweets to create two clusters – English, and non-English tweets. This approach relies heavily on the text contained in the tweets.

In this paper, we suggest ways for using unsupervised methods to avoid the task of manually labeling a huge amount of tweets. A similar problem was also tackled by Nigam et al. (2000) in which they suggest a combination of Naive Bayes and EM to classify text. Using this method, we need very little labeled data in order to classify large amounts of unlabeled text. Our method does not need any labeled text and is based on wordlists of languages and a simple heuristic. It relies on the fact that huge amounts of text would eventually help in converging to the correct model, with negligible errors. There are abundant tweets available via the Twitter API, and hence this method works very well.

3 High-confidence dictionary-based labeling algorithm

As mentioned earlier, getting labeled twitter data is not a trivial task. All methods which were based on training data used manually labeled tweets for their learning algorithms. However, this might restrict the number of tweets used for training to a few thou-

²<http://blog.echen.me/2011/05/01/unsupervised-language-detection-algorithms>

Language	Number of tweets labeled
English	1157034
Spanish	772014
German	20548
French	48766
Dutch	142666

Table 1: Number of tweets used for obtaining language models

sands. The language identification algorithm could perform better if it is trained using a lot more tweets. For this, an unsupervised way of getting labeled data is proposed. The method is quite simple – use words from dictionary of languages and count the number of valid words and the proportion of such valid words from each language that has appeared in a tweet. If these values for a language L in the tweet exceeds a certain threshold, then it can be labeled with L . We used Wiktionary, which is a freely available source of wordlists, for building wordlists for languages. Initial experiments showed that the unsupervised labeling algorithm achieves more than 90% accuracy when the threshold for the number of valid words is set to 4 and proportion of valid words was set to 0.6. If the criterion is not met, the tweet is ignored. In this way, a lot of accurately labeled tweets can be collected. Even though all tweets might not be labeled, the fact that the labeling operation is unsupervised gives a powerful method to acquire large amounts of labeled training data. Also, slight inaccuracies in identifying the labels using this approach are negligible, as the number of tweets is very high.

The above method was run on a dataset containing more than 10 million tweets, and the number of tweets that were labeled by the above algorithm is tabulated in Table 1.

The first step is to acquire labeled tweets using the technique described above. Also, currently, the languages that would be considered are – English, Spanish, German, French, Dutch and Unknown. If enough words are not found from these languages in a tweet, it is ignored. A tweet is labeled ‘Unknown’ if the number of words in the tweet that does not match any of the known dictionary words exceeds a certain threshold. For example, if a tweet contains 10 words, out of which only 1 word is recognizable as English, then it means that the tweet contains 90% unknown words and hence labeled ‘Unknown’.

Once a set of labeled tweets is obtained, a language model is built for each language using this training data. These are built using character n-gram frequencies. Even though there have been approaches based on character n-grams, our approach is different in the sense that – the data is obtained automatically from twitter, and a huge amount of tweets will be used for generating the language models, without using formal text from any other sources. We expect this to be a better representative of the actual tweets, and hence produce better results.

4 Language detection

For classifying an unlabeled tweet, the character n-grams of the tweet were generated (where n ranges from 1 to 5), and the correct language was identified using TextCat, a text categorization tool. In order to generate a model that is compatible with TextCat, the n-grams were sorted in descending order of their counts. Also, not all the n-grams were included in the language model. Top 500 character n-grams were taken into consideration because there is a possibility of error as the counts of n-grams decreases. Initial experiments show that top 500 n-grams give best results (Figure 2).

TextCat performs language detection by generating n-grams of the input text and comparing it with the language models on which it has been trained. For our experiments we used the models that we obtained using our unsupervised labeling method. TextCat assigns a *distance* or *score* for each language, and the predicted language is the one corresponding to the least distance from the language model.

In order to identify languages more accurately, the *blogger prior* was also calculated for the author of each tweet as in Carter et al. (2012). This was done by obtaining the last few tweets (around 150-200) of an author, and calculating the scores for each of the five languages. This language prediction task of the tweets of bloggers was also done using TextCat trained on our models. Let $T = \{t_1, t_2, \dots, t_k\}$ be the previous k tweets of a user/blogger. The blogger profile is calculated as follows : for each tweet $t_i \in T$, find the scores calculated by TextCat for each language L . The

Dataset	Number of tweets
Development	401
Training	2003
Testing	1987

Table 2: Number of tweets in the dataset provided by Carter et al (2012)

blogger profile is then the vector (s_1, s_2, \dots, s_l) where l is the number of languages over which TextCat is trained, and s_i is the average score calculated by TextCat for language i over all the k tweets. Once the blogger prior is calculated, it is combined with the prediction for actual tweet using a weight factor w as follows:

$$\lambda_{combined} = w * \lambda_{blogger} + (1 - w) * \lambda_{tweet} \quad (1)$$

As mentioned in Carter et al. (2012), including the blogger prior does improve the accuracy of language prediction task. It was seen that with this single additional prior, overall accuracy of over 96% was achieved by our language models. More details are discussed in the results section.

5 Data set

This section describes the dataset that was used for carrying out the experiments. Carter et al. (2012) have provided a development, training and testing set for the five languages – English, Spanish, French, Dutch and German. We used their development set for tuning our parameters (such as number and proportion of valid words, the weight factor for combining tweet and blogger profiles). For the results that are reported in this paper, their testing set was used. The breakup of the number of tweets in development, training and testing set in their dataset is given in Table 2. It should be noted that some tweets were not available due to certain reasons (users protected their tweets, or deleted their accounts). Hence the number of tweets that were used in the actual work by Carter et al (2012) is different from what was used in our work.

The number of tweets that were labeled (the ones that were used to build the language model) for each language is tabulated in Table 1.

6 Results

In this section, the results of certain experiments are reported. The first step in proposed approach is to identify the language of tweets based on the number (and the proportion) of valid words. Some sample data ³ was taken, and this method of labeling was employed on that sample set. The dataset contains 401 tweets, which have been labeled manually with the language. For this experiment, the labels were removed, and the tweets were fed as input to our algorithm.

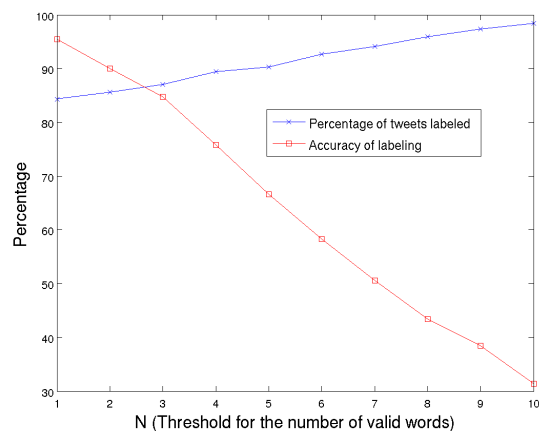


Figure 1: Accuracy of labeling method based on number of valid words in a tweet

Figure 1 shows the overall accuracy of the labeling algorithm along with various values for the threshold for minimum number of valid words in a tweet required for classification. It also shows the number of tweets that were actually classified, that is, the number of tweets that were not ignored. As expected, the accuracy of labeling increases as N increases (the smooth line with 'X's, in Figure 1). We can observe that at $N=4$, the accuracy is above 89%, and that more than 75% of tweets were labeled. As N increases, the number of labeled tweets falls rapidly (line with squares, in Figure 1). This experiment serves as a proof of concept for the effectiveness of labeling based on number of valid words in a tweet. The results are promising, and a large number of labeled tweets were obtained using this method.

³The dataset was taken from the development data used in <http://ilps.science.uva.nl/resources/twitterlid>