Bootstrapping Websites for Classification of Organization Names on Twitter

Paul Kalmar

Kalmar Research
paul@KalmarResearch.com

Abstract. There has been a growing interest in monitoring the social media presence of companies for improved marketing. Many public APIs are available for tapping into the data, and there are companies that will collect all posts related to a given set of keywords. But with so much data, who is to say that all of the posts are relevant, especially when so many company and product names are highly ambiguous? In the context of the WePS Task 2, we aim to reduce noise by collecting only the relevant tweets about a company given the company's website and set of Twitter data. In a real world situation, any company who wanted to identify tweets about themselves could provide a short list of labeled tweets and use this as a base set for training data. Given that for this task we were given a large list of companies with no such training data, it would have been unrealistic to create such data for each company. We chose to use the company's website as surrogate training data. Because the websites come from a different register than Twitter, we used the initial model to bootstrap a model from the actual tweets. As it is the most simple data to acquire, the features we chose to use were the co-occurring words in each tweet. To compute the relevance of each word to a given company, we computed the pointwise mutual information between the word and the target's label. The results show that our approach was successful, yet with room for improvement.

Keywords: bootstrap, unsupervised, Twitter, disambiguation

1 Introduction

There has been a growing interest in monitoring the social media presence of companies for improved marketing. Many public APIs are available for tapping into the data, and there are companies that will collect all posts related to a given set of keywords. But with so much data, who is to say that all of the posts are relevant, especially when so many company and product names are highly ambiguous? In the context of the WePS Task 2, we aim to reduce noise by collecting only the relevant tweets about a company given the company's website and set of Twitter data.

2 Method

For the task of classification, there needs to be at least one well defined class. In a real world situation, any company who wanted to identify tweets about themselves could provide a short list of labeled tweets and use this as a base set for training data. Given that for this task we were given a large list of companies with no such training data, it would have been unrealistic to create such data for each company. We chose to use the company's website as surrogate training data. All text was extracted from the site and used to create an initial model. Meta tags such as keywords and description were heavily weighted.

2.1 Data collection

To collect the data, we used Python to query and cache each webpage listed for the data sets. The cached page was then processed using BeautifulSoup, extracting all text and meta tags. Some website addresses were modified from the original data set to potentially get a more correct or textual site. This was done manually, but in future research a process that warns about pages with low amounts of text would be preferred. One of the main changes was correcting Wikipedia page addresses to the English form of the page and collecting the raw text.

2.2 Features

As it is the most simple data to acquire, the features we chose to use were the co-occurring words in each tweet. To compute the relevance of each word to a given company, we computed the pointwise mutual information between the word and the target's label. Pointwise mutual information is defined as the log of the probability of the word occurring with the label divided by the product of the probability of the word and the probability of the label.

$$PMI = \log(p(word,label) / (p(word)*p(label)))$$

2.3 Normalization

Originally we planned to use the data completely unnormalized for various reasons. The primary reason is that the less is done to alter the starting data, the more reliably the system would perform in a completely unsupervised setting with unknown data. Twitter has many terms and spellings which distinguish the language used there from standard English. Also, the original specifications for the task included Spanish language data, which would have either required an additional Spanish system or, as we opted for, a language agnostic system.

Given that the task eliminated non-English data, we added some simple normalization to the text. All words were converted to lowercase and passed through a stemmer from NLTK. [1]

2.4 Bootstrapping

Because the company websites come from a different register than Twitter, we used the initial model to bootstrap a model from the actual tweets. We examined multiple approaches to bootstrapping the model. To build the bootstrapped model, we took the initial model built from the company website and applied it to label the company's tweets. The set of tweets that the initial model was most confident in labeling were then taken and used to build subsequent models. This bootstrapped model was then incremented over several iterations by repeating the process and retaining a larger set of tweets. There are three variables that had to be determined: size of initial bootstrapped model, method of incrementing model, and number of iterations.

Size of model. Optimally, the initial bootstrapped model should have all correctly labeled tweets and none of the incorrectly labeled tweets. As that is highly unlikely, it is best to err on the side of precision as opposed to recall. To determine the size of the model, two possible options are thresholding and limiting the set to a fixed number or percentage of tweets. We developed models using two different methods: 1. Keep all tweets with confidence greater than or equal to the median confidence. 2. Keep a specific percentage of best confidence labeled tweets. For future research, it might be interesting to use machine learning to determine the optimal confidence threshold and use this.

Method of incrementing the model. Once the initial model is created, we iteratively increment the model on the set of tweets. To do so it is necessary to again determine how much more data to add, and whether to include the previous data in the model. With the initial model which relied on the median confidence, we stuck with this method as we iterated through the tweets. For the model that kept a specific percentage of best confidence tweets, we gradually increased this percentage by a small amount each iteration up to a maximum size.

Number of iterations. The last variable to choose for bootstrapping was the number of iterations to perform. For future research, the optimal stopping point could be computed with machine learning. Due to time constraints, we tried varying the number and looking at the results.

Submitted configurations.

For the WePS task, we submitted results from the following configurations.

Table 1. Submitted configurations.

Configuration Name	**Initial model size**	**Increment percentage**	**Number of iterations**
KalmarResearch_1*	Median	Median	40
KalmarResearch_2	Median	Median	40
KalmarResearch_3	Median	Median	35
KalmarResearch_4	10.00%	2.00%	30
KalmarResearch_5	15.00%	1.00%	25

**KalmarResearch_1 was a slight variation in the code from KalmarResearch_2*

3 Evaluation

This task consisted of a binary classification of tweets, and therefore can be evaluated using standard classification metrics. Some such metrics are accuracy, precision, recall, and f-measure which is the harmonic mean of precision and recall.

Table 2. Metrics.

Accuracy	Precision	Recall	F-Measure
(CP+CN)/ (CP+FP+CN+FN)	CP/(CP+FP)	CP/(CP+FN)	(2*Precision*Recall)/ (Precision + Recall)

CP = #Correct Positives, FP = # False Positives, CN = #Correct Negatives, FN = #False Negatives

Of these, accuracy reflects performance across all classes whereas the others reflect performance with regard to a specific class. Accuracy is the only one of the above metrics that utilizes the number of correct negatives.

Although the evaluation metric used for the WePS task was accuracy, we chose to focus on the f-measure for the positive class. When using a system such as this, finding negative results is not in any way helpful -- what matters are the results that are actually about the company. Recall is important so that all possible results are returned about the company, and precision is important so there are few to no false positives in the results. We realized early on that accuracy would be of little actual use to us, so we did not attempt for high accuracy results. Instead, we focus on results for the harmonic mean of precision and recall of positive results.

4 Results

The following table is the official results from the WePS task for our systems.

SYSTEM	Accuracy	# not answered	Precision (positive)	Recall (positive)	F-Measure (positive)	Precision (negative)	Recall (negative)	F-Measure (negative)
KALMAR_1	0.4	2207	0.51	0.7	0.42	0.59	0.19	0.21
KALMAR_2	0.44	874	0.47	0.7	0.43	0.61	0.27	0.28
KALMAR_3	0.4	2202	0.49	0.66	0.39	0.66	0.25	0.27
KALMAR_4	0.46	874	0.48	0.75	0.47	0.65	0.25	0.28
KALMAR_5	0.44	874	0.48	0.77	0.47	0.65	0.21	0.23

The results show that our approach was successful, yet with room for improvement. Many sites contained little or no text, which caused our approach to be ineffective for these companies. As expected, the system which used the smallest initial model and a large number of iterations generally performed the best.

5 Discussion

Because our approach was based on co-occurring words, the best results appear when the keyword is disambiguated to its correct sense. Many of the companies used the keyword in the same sense as non-company related tweets, however, and this caused a high error rate in our system. An example of this is sports teams, where the name of the city is used in the same sense in the name of the team and when referring to the city in general. For keywords which had a completely different meaning than the company name, results were much more accurate.

6 Conclusion

Although our approach achieved lower results than expected, this seems to be a good initial pass which can be improved by automatically setting variables with machine learning. This system achieves high results on labeled training data, and is therefore a suitable approach for normal supervised scenarios.

7 References

1. Bird, S., Loper, E., and Klein, E.: Natural Language Processing with Python. O'Reilly Media Inc (2009)

2. Kalmar, P.: Less is More: Advantages of Using Local Homogenous Data Sets in Natural Language Processing. Master's Thesis at San Diego State University (2008)