Developing a Hierarchical Multi-Label Classifier for Twitter Trending Topics

Jinan Fiaidhi[1], Sabah Mohammed[1], Aminul Islam[1], Simon Fong[2] and Tai-hoon Kim[3]

[1]Department of Computer Science
Lakehead University, Thunder Bay, Ontario P7B 5E1, Canada
[2]Faculty of Science and Technology, University of Macau, Macau, China
[3]Department of Computer Engineering, Glocal Campus, Konkuk University, Korea
[1]{jfiaidhi, mohammed, maislam}@lakeheadu.ca, [2]ccfong@umac.mo,
[3]taihoonn@kku.ac.kr

Abstract

In recently years, there has been rapid growth in discussion groups and micro blogging, in which an important characteristic of the entries is their trending topics on some generalized categories. Many researchers have attempted to classify trending topics by using only keywords, trending topics are rarely straightforward; they are normally expressed in a more subtle manner. It is well accepted that using high-dimensional multi-modal language features for tweets content representation and classifier training may achieve more sufficient characterization of the diverse properties of the tweets and further result in higher discrimination power of the classifiers. However, training the classifiers in a high-dimensional multi-modal feature space requires a large number of labeled training tweets, which will further result in the problem of curse of dimensionality. To tackle this problem, a hierarchical feature subset selection algorithm need to be used to enable more accurate tweets classification; where the processes for feature selection and classifier training are seamlessly integrated in a single framework. In this article, we used the LingPipe classifier to accurately classify the Twitter trending topics where it shows a substantial improvement over their state-of-the art trending topics-trained counterparts.

Keywords: *Trending topics; Trending Topics Classification; LingPipe API*

1. Introduction

Currently Twitter is heavily used as a source of communication. People are busy writing on Twitter about what's going around and within their personal space as well as on variety of shared issues. With the torrential streams of tweets, there's an emerging demand to sieve signals from noises and harvest useful information. Besides Twitter Search[1] , there are many Twitter Analytics tools (*e.g.*, TwitAnalyzer[2] , MicroPlaza[3] , Twist [4], TwitTruly[5] , TweetStats[6],

[1] https://twitter.com/search
[2] http://www.twitalyzer.com/5/index.asp
[3] https://twitter.com/microplaza
[4] http://techcrunch.com/2008/04/23/twitter-trends-twist/
[5] http://twitturly.com/
[6] http://www.tweetstats.com/

TwitterFriends[7]) to analyze Twitter streams. Each of these tools serves specific purpose. They crawl and sift through Twitter streams; also, aggregate, rank and slice-and-dice data to deliver some insights on Twitter activities and trends. There's no single best analytic tool available but for some cases a combination of these tools may extract interesting insights from Twitter streams [1]. Similarly, the tools for analyzing popular topics or trending topics on Twitter (*e.g.*, Datasift[8] , What the Trend[9] , Trendsmap[10]) fails to accurately classifying tweets based on general categories [11]. Even Twitter allows users to observe only limited number of trending topics where these topics are restricted to the top ten popular terms of discussion at any given moment.

Due to the volume of tweets, it is natural to consider techniques like named-entity recognition, information extraction, and text mining over tweets. Not surprisingly, the performance of "off the shelf" natural language processing tools, which were trained on news corpora, is weak on tweet corpora [2]. To address this challenge we need to develop a technique that can identify types of the entities tweets may contain. This paper presents a new methodology, which helps improve the accuracy of tweets trending topics classification. Unlike most prior works which focused on lexical features at the word level, the methodology presented here attempts to include more contextual information by focusing on the whole tweet level by using a classifier that takes into account the language model. This is a dynamic classifier that accepts training events of categorized character sequences based on a multivariate estimator for the category distribution and dynamic language models for the per-category character sequence estimators. Luckily, we do not need to describe and implement this type Classifier[11] has been developed by a research group at Carnegie of dynamic classifier as the LingPipe's LanguageModel (LM) Mellon University where they provided a suite of Java libraries for the linguistic analysis of human language. Using the LingPipe classifier, we managed to more accurately classify the Twitter trending topics where it shows a substantial improvement over their state-of-the art trending topics-trained counterparts.

2. Related Research

Twitter, a popular micro-blogging site that present opportunities for research in natural language processing (NLP) and machine learning. One of such opportunities is trending topics classification [3]. There are number of research papers addressed twitter classification, sentiment classification as well as on trending topics classification. James Bernhard's [4] classifies the text to a predefined set of generic classes such as News, Events, Opinions, Deals, and Private Messages based on domain-specific features extracted from the user profile and text. Alec Go [5] introduced a method to automatically classify of Twitter messages either positive or negative. In this direction they used machine learning algorithms for classifying the sentiment of Twitter messages using distant supervision. Sheila Kinsella

[7] http://stats.brandtweet.com/
[8] http://datasift.com/
[9] http://whatthetrend.com/
[10] http://trendsmap.com/
[11] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/listen/NLP/Parsers/Stanford/stanford-parser-2005-07-21/MCALL/lingpipe-3.6.0/index.html

[6] included external hyperlinks and metadata to classify Social Media data. She showed that external metadata has better descriptive power for topic classification than the original posts and gives better classification results. Sankaranarayanan [7] have introduced a method that can be used to automatically obtain breaking news from the tweets. In particular, this article uses noisy data along with a naïve Bayesian classier to improve the quality of the noisy data by throwing away a large portion of the tweets noise. Backer [8], however, introduced an approach to distinguish between real-world events from a family of non-events messages. He used an online clustering technique that groups together topically similar tweets, and computed features that can be used to train a classifier to distinguish between event and non-event clusters. Arkaitz Zubiaga [9] introduces a typology to categorize the triggers that leverage trending topics: news, current events, memes, and commemoratives by defining a set of straightforward language-independent features that rely on the social spread of the trends to discriminate among those types of trending topics. Thongsuk [10] proposed a framework for classification by using Twitter posts from three business types, *i.e.*, airline, food and computer & technology. They used feature transformation and feature expansion to classify business type tweets on Twitter. K Lee [11] classifies Twitter Trending Topics into 18 general categories. They used Bag-of-Words approach for text classification and network-based classification. In twellow[12] they collected publicly available messages and scans users profiles from the Twitter.com in order to categorize users and identify those users responsible for those messages into the various categories. However, none of these research attempts try to classify Twitter trending topics based on personalized attributes.

3. Identifying and Categorizing Trending Topics

In our preparation to categorize and classify Twitter trending topics we collected a reasonable tweets dataset using the Twitter streaming API[13] , with the filter tweet stream providing the input data and the trends/location stream providing the list of terms identified by Twitter as trending topics. The filter streaming API is a limited stream returns public statuses that match one or more filter predicates. The United States (New York) and Canada (Toronto) was used as the location for evaluation. Google Geocoding API[14] has been used to get location wise Twitter data. The streaming data was collected automatically using the Twitter4j API[15]. The streaming data was stored in a tabular CSV formatted file. Data has been collected with different time interval for the same city and topics. In this direction, we have collected different topics dataset for different city with different time interval. For Labeling we identified 12 general classes for topic classification. These classes are Politics, Education, Health, Marketing, Music, News & Media, Recreation & Sports, Computers & Technology, Pets, Food, Family, and other. Since twitter is our primary source we have used the twitter search API to search topics and manually label the topics. If the tweets are related to political issues then they will be classified as politics. If the topic is not related to any Category then

[12] http://www.twellow.com
[13] twitter.com
[14] developers.google.com/maps/documentation/geocoding
[15] Twitter4j.org

the topic will be classified as other category. The distribution of collected data over the 12 classes is provided in Figure 1.

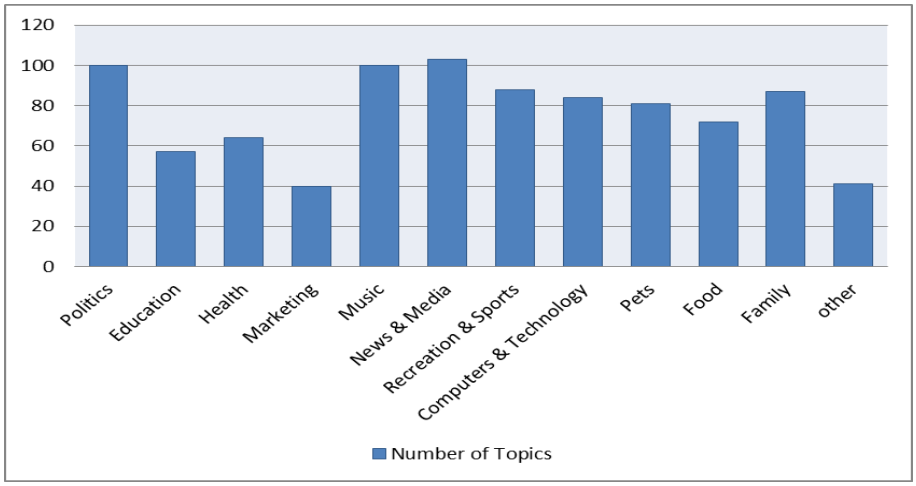


Figure 1. Label of 917 Topics Across 12 Classes

Having collected a reasonable dataset for generally identified tweets across the 12 general classes, we can then use this dataset to training effective text classifiers for classifying newly collected tweets. Indeed machine learning [12] can be used as a general inductive process to automatically builds a text classifier by learning the characteristics of a set of previously classified documents. These characteristics are then used to classify new documents. Different types of Text Classification tasks can be defined [13] between single-label and multi-label classification. In Single-label (also called multi-class) Text Classification, exactly one category must be assigned to a document. In Multi-label Text Classification, any number of categories may be assigned to a document. Binary categorization is a special case of single-label categorization, in which there is only one category and each document can be assigned to it or not. Many classification methods, such as Naïve Bayes, Support Vector Machine, are of the single-label type. The most popular approach for multi-label classification is binary approach [14] but this method has two main problems. First, it assumes independence of categories, which is not always true and second problem is that a big number of binary classifiers have to be learned, which may cause memory problems, and take a lot of time. Hierarchical classification [15-16] has advantages compared to flat classification it enables easy location of required categories which makes it easier to search among large number of categories and sub categories. It also reflects the intuition of relatedness of topics that are close to each other in the hierarchy. Two hierarchical classification methods big-bang and top-down level based approach. In the big-bang approach, a document is classified into a category in the category tree by a classifier in one single step. In the top-down level-based approach, one or more classifiers are constructed at each level of the category tree, and each classifier works as a flat classifier at that level [14]. Koller [16] divide the hierarchical classification task into a set of smaller classification tasks, each of which corresponds to some split in the classification hierarchy. In their result the size of the classifier allow to obtain significantly higher accuracy, a reduction due both to increased robustness and to our ability to use more accurate classifiers. Figure 2 shows our Trendy Topics Classification approach.

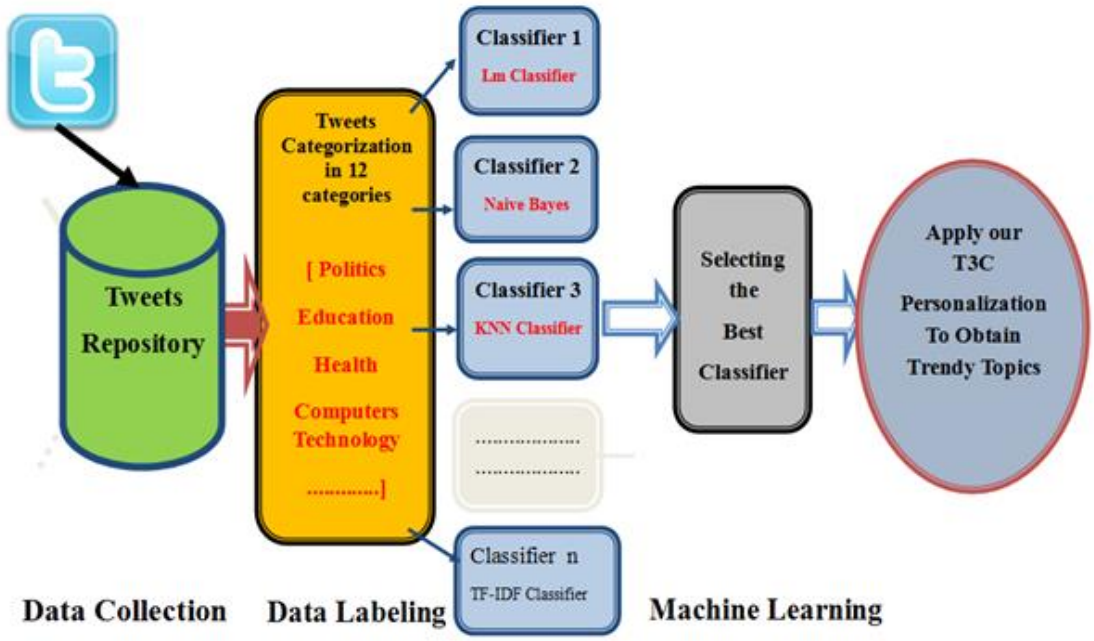


Figure 2. Twitter Trendy Topics Classification Structure

We have implemented hierarchical multi-label classification algorithm using a flat multi-class classifier provided by LingPipe API16 . LingPipe's LanguageModel (LM) Classifier performs joint probability-based classification of character sequences into non-overlapping categories based on language models for each category and a multivariate distribution over categories. The LingPipe's LM classifier is a language model classifier that accepts training events of categorized character sequences. Training is based on a multivariate estimator for the category distribution and dynamic language models for the per-category character sequence estimators. It calculates conditional and joint probabilities of each category for the classified object. A scoring classifier goes one step further and assigns a (floating point) score to each category. These may then be sorted to provide a ranking and a first-best result, with higher scores taken to be better matches and LingPipe classifier returns one best category as result of classification process. For multi-label classification, we apply an approach based on estimations of probabilities of an item to belong to some category. To determine the threshold for multi-label classification we use the cross-entropy scores provided by LingPipe classifier, as they are better suited for cross document comparison.

A Naive Bayes classifier assumes that the presence of a particular feature of a class is unrelated to the presence of any other feature given the class variable[17] . A classifier is constructed from a set of categories and a tokenizer factory. For this purpose we have used Whitespace Tokenizer Factory. Naive Bayes applied to tokenized text results in a so-called "bag of words" model where the tokens (words) are assumed to be independent of one another[18] . This classifier has been implemented as NaiveBayesClassifier class in Lingpipe direct derivative of the DynamicLMClassifier as per LingPipe API doc.

The K Nearest Neighbor (Knn) Classifier uses the K Nearest Neighbor algorithm to classify Data. A KnnClassifier implements k-nearest-neighbor classification based on feature

[16] http://alias-i.com/lingpipe/
[17] http://en.wikipedia.org/wiki/Naive_Bayes_classifier
[18] http://alias-i.com/lingpipe/docs/api/index.html

extraction and a vector proximity or distance. K-nearest-neighbor classification is a kind of memory-based learning in which every training instance is stored along with its category . In the training phase the algorithm stores feature vectors of the training examples along with their categories[19]. The features are extracted with the "bag of words" model using Whitespace Tokenizer Factory. This classifier has been implemented using LingPipe KnnClassifier class.

The TF-lDF Classifier is based on term frequency and inverse document frequency to classify data. LingPipe's TF-IDF classifier training phase is similar to that used for the Knn and Naive Bayes classifiers. The features are extracted with the "bag of words" model using Whitespace Tokenizer Factory. This classifier has been implemented using LingPipe TfIdfClassifier class.The process of selecting the best of these classifiers can illustrated using the following code snippet:

```
var classifier = DynamicLMClassifier.createNGramProcess (CATEGORIES,Ngram_Size);
      for(int i=0; i<CATEGORIES.length; ++i) {
         var Dir = getCategoryDirectory();
         var files = getListofFile(Dir);
         for (int j = 0; j < files.length; ++j) {
            var text = read(Dir,files[j]);
          text = applyWordNetSynonym(text) ;
var classification= new Classification(CATEGORIES[i]);
var classified= new Classified (text,classification);
            classifier.handle(classified);    }
}
var compiledC= AbstractExternalizable.compile(classifier);
  var evaluator = new ClassifierEvaluator<> (compiledClassifier,  CATEGORIES,
storeCategories);
      var _listOFFiles = Directory.listFiles();
       for(int i = 0; i < CATEGORIES.length; ++i) {
          for(int k = 0; k < _listOFFiles.length; ++k){
               var text = readFile(_listOFFiles[k]);
var classification      = new Classification(CATEGORIES[i]);
var classified = new Classified (text,classification);
               evaluator.handle(classified);
              var jc = compiledClassifier.classify(text);
            String bestCategory = jc.bestCategory();}
       }
var summery = evaluator.confusionMatrix().microAverage();
for(c=0; c<CATEGORIES.length; ++c) {
var catSummery = evaluator.oneVersusAll(CATEGORIES[i])
}
```

First it initializes the classifier with category array and the n-gram size then the loop continues through the categories. The training data is organized into directories by category, and then the training files are read from the file using the LingPipe utility method. After that we applied the Word NetSynonym[20] database to get synonym for each the tweet word wherever possible. The resulting data is used to train the classifier for the specified category.

[19] http://alias-i.com/lingpipe/docs/api/com/aliasi/classify/KnnClassifier.html
[20] http://wordnet.princeton.edu/wordnet

Then it creates an evaluator from the classifier. Next for each category we have read all testing data and execute the provided LingPipe classifiers to get best category according to given training dataset. This will continue until the end of all categories. We repeat the same process for each category because each testing dataset can be assign to multiple categories. After classified the dataset into 12 different categories we then apply our T3C [17] method to personalize trending topics. This iterative process will return at the end the summery of testing result sets for each category.

4. Experiments and Results

Our experimentation starts by collecting reasonable tweets samples on general topics like health, education, sports, economy, Family, Technology, Music and politics. First we collected random Tweets using Twitter Streaming API. For Labeling we build an Interface to label data into 12 different categories[21]. We have labeled two different dataset to experiment our result. We have collected tweets using the Twitter Streaming API and label them into 12 different category and for second dataset we have apply T3C [17] to get trending topics and then labeled the topics. During labeling process tweets were preprocessed to remove URL's, Unicode characters, usernames, and punctuation, html, etc. A stop word file[22] containing common English stop words was used to filter out tweets from common words. For First experiment we have collected 100,000[23] Tweets. Language Model (LM) Classifier, Naive Bayes Classifier, K-Nearest Neighbor (Knn) Classifier, and TF-IDF Classifier were chosen for the experiment. Table 1 presents our results sets where we use the overall classifier accuracy for the classifier performance. Figure 3 shows the performance comparison graph

Table 1. Performance for Lingpipe Classification Experiment

	Language Model Classifier			Naïve Bayes Classifier			K-Nearest Neighbor			TF-IDF Classifier		
Category	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision
Politics	0.8	0.14	0.08	0.8	0.8	0.14	0.85	0.8	0.14	0.82	0.12	0.08
Education	0.88	0.05	0.08	0.88	0.88	0.05	0.51	0.88	0.05	0.88	0.05	0.08
Other	0.86	0.07	0.08	0.86	0.86	0.07	0.91	0.86	0.07	0.85	0.08	0.08
Health	0.83	0.1	0.08	0.84	0.83	0.1	0.9	0.83	0.1	0.88	0.05	0.08
Marketing	0.88	0.05	0.08	0.87	0.88	0.05	0.88	0.88	0.05	0.87	0.05	0.08
Music	0.84	0.09	0.08	0.84	0.84	0.09	0.84	0.84	0.09	0.86	0.07	0.08
News & Media	0.82	0.11	0.08	0.81	0.82	0.11	0.86	0.82	0.11	0.83	0.11	0.08
Recreation & Sports	0.83	0.11	0.08	0.82	0.83	0.11	0.85	0.83	0.11	0.81	0.13	0.08
Computers Technology	0.86	0.07	0.08	0.86	0.86	0.07	0.9	0.86	0.07	0.84	0.09	0.08
Pets	0.86	0.07	0.08	0.86	0.86	0.07	0.91	0.86	0.07	0.86	0.07	0.08
Food	0.84	0.09	0.08	0.84	0.84	0.09	0.85	0.84	0.09	0.82	0.12	0.08
Family	0.88	0.04	0.08	0.88	0.88	0.04	0.9	0.88	0.04	0.86	0.07	0.08

[21] http://flash.lakeheadu.ca/~maislam/Mining-DataSet/TrainedData/
[22] http://flash.lakeheadu.ca/~maislam/Data/stopwords.txt
[23] http://flash.lakeheadu.ca/~maislam/Mining-DataSet/TestingData/

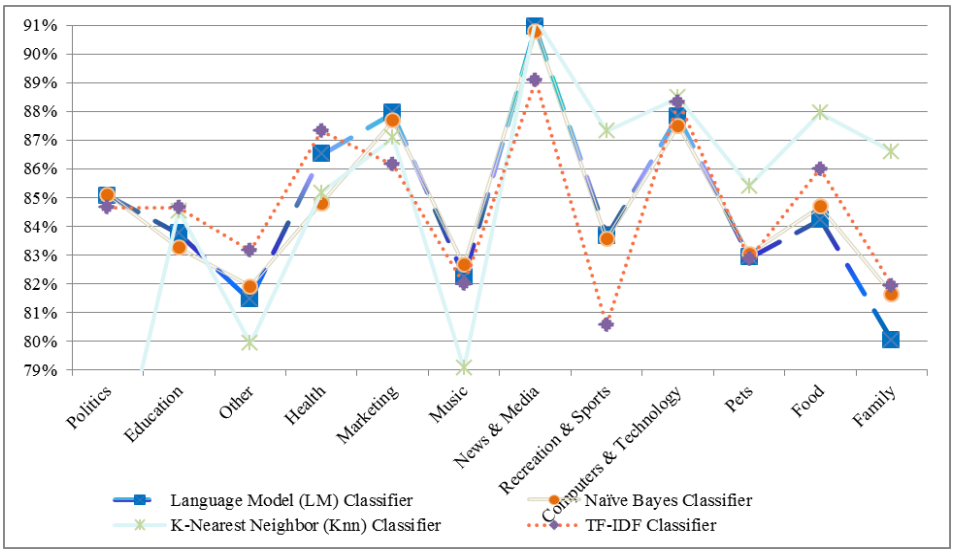


Figure 3. Comparison Graph for Lingpipe 4 Classifier Results

For the Language Model (LM) Classifier algorithm, the size of the n-gram needs to be set. N- gram is a sub-sequence of length n of the items given. The Language Model rule is to classify a newly given document based on prediction occurring n-grams.

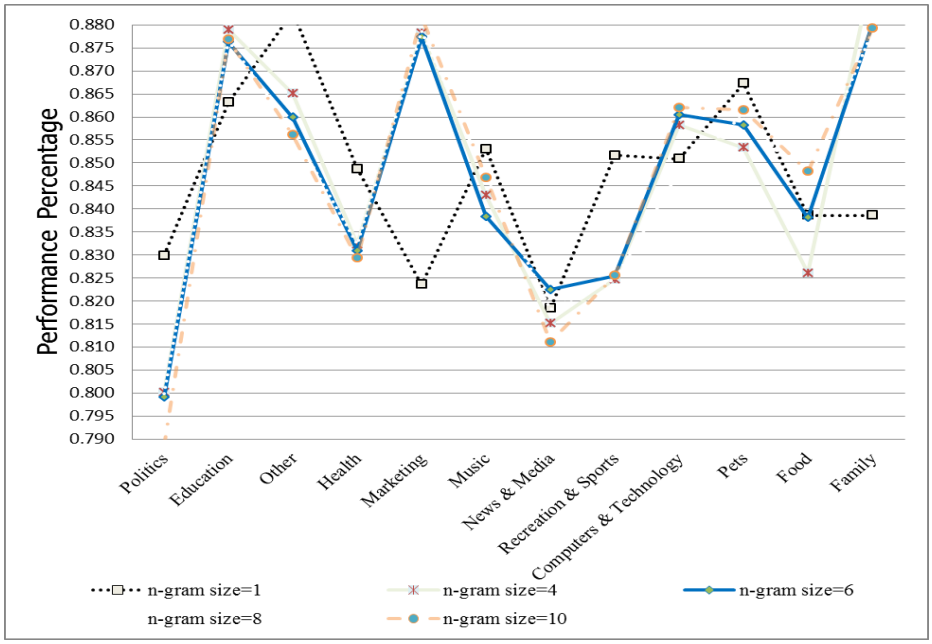
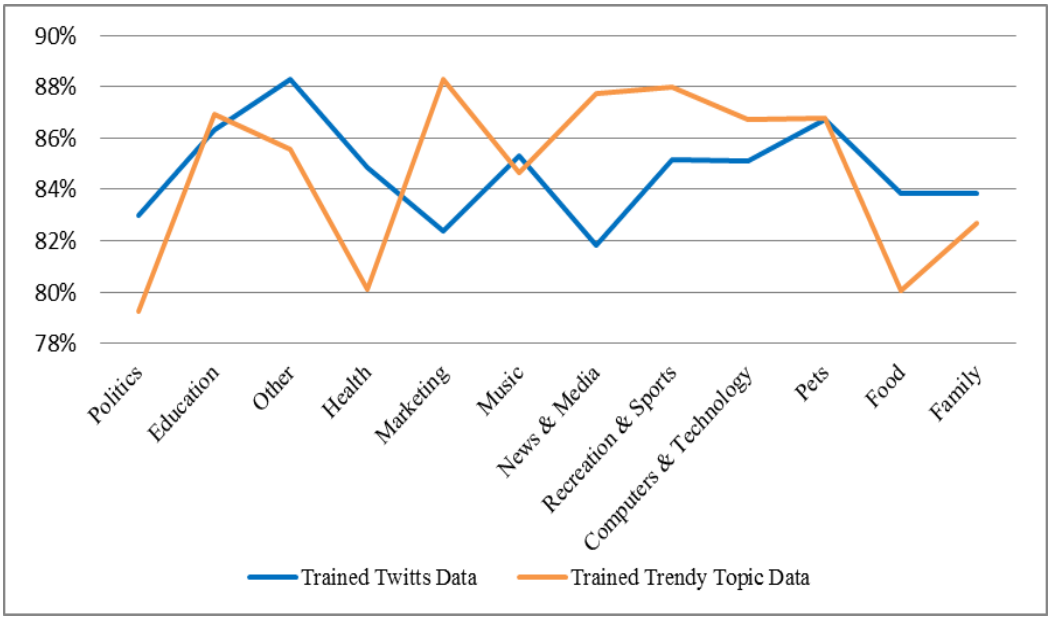


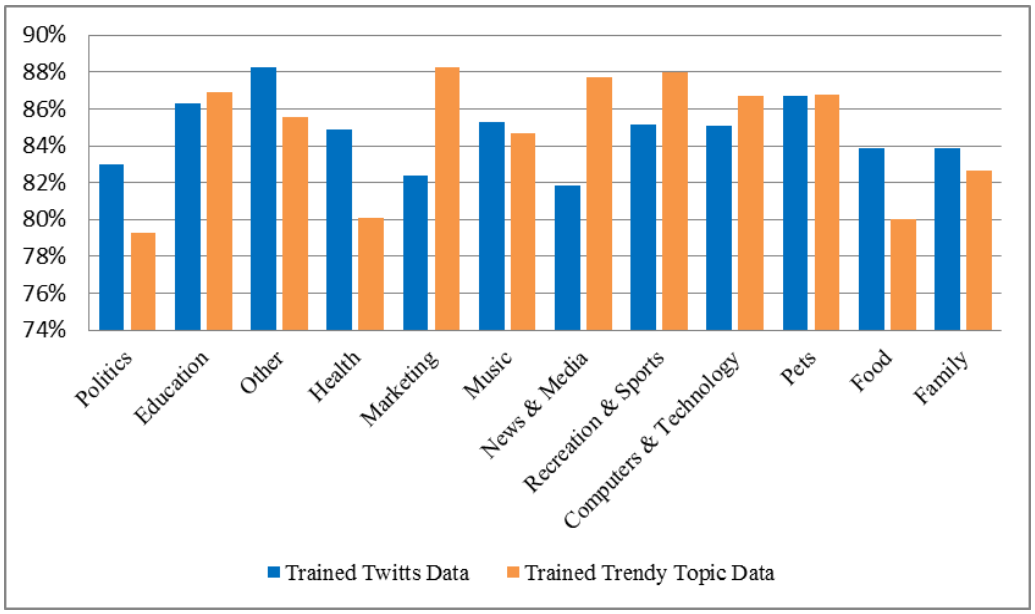
Figure 4. Performance Graph for n-gram Size LM Classifier

This algorithm uses a character based n-gram to classify Tweets so an appropriate size should be the average length of a word. Figure 4 shows performance Graph for n-gram Size Language Model (LM) Classifier.

Figure 5 show the overall performance accuracy Comparisons graph when we apply Lingpipe Classification algorithm on Twitter Trendy Topics dataset and General Tweets dataset.



(a) Comparison Graph



(b) Comparison Histogram

Figure 5. Comparing Trendy Topics Categorization based on Two Different Trained Dataset

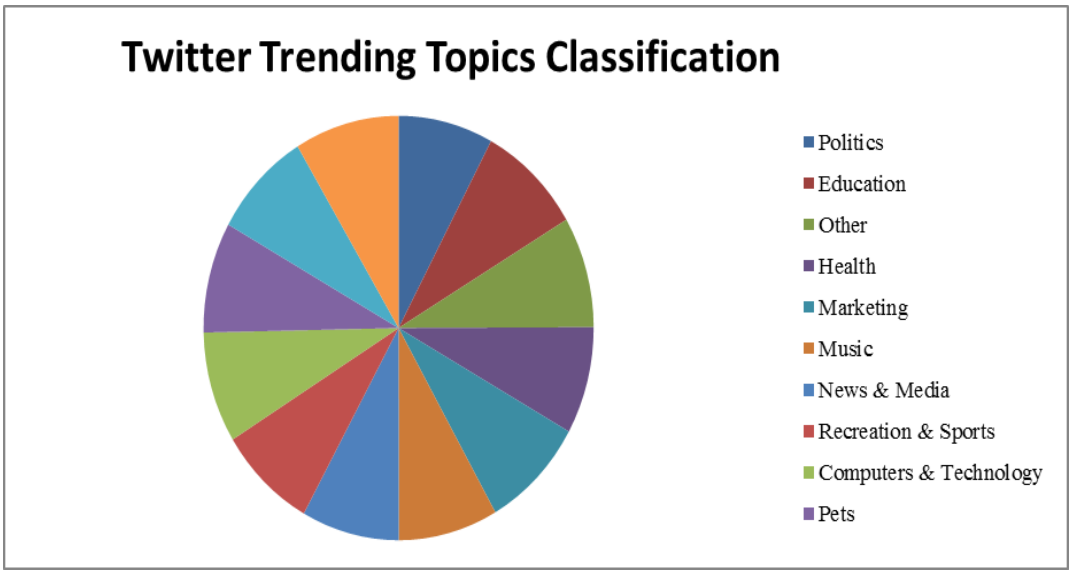


Figure 6. Diagram for Twitter Trending Topics Classification

5. Conclusion

In this article we used a high-dimensional multi-modal language features for tweets content representation and classifier training to accurately characterizing the diverse properties of the tweets and further result in higher discrimination power of the classifiers. However, training the classifiers in a high-dimensional multi-modal feature space requires a large number of labeled training tweets, which will further result in the problem of curse of dimensionality. To tackle this problem, a hierarchical feature subset selection algorithm need to be used to enable more accurate tweets classification; where the processes for feature selection and classifier training are seamlessly integrated in a single framework. For this purpose we have applied four supervised machine learning algorithms Language Model (LM) Classifier, Naive Bayes Classifier, K-Nearest Neighbor (Knn) Classifier, and TF-IDF Classifier. All the results of these experiments were published at our Lakehead University Flash server[24] . We found that well trained machine learning algorithms can provides very good classifications to the Twitter Trending Topics. In terms of overall performance accuracy, all four algorithms can reach more than 75% of classification correctly. However, the Language Model (LM) Classifier in N-gram model performs better than the other three classification algorithm. Also our experiment show that a larger twitter training data set perform better in Trending Topic classifications over trending topics training dataset. Figure 6 shows the Multi-Label Twitter Trending Topics Classification diagram. For this purpose, we used the LingPipe classifier to classify the Twitter trending topics where it shows a substantial improvement over their state-of-the art trending topics-trained counterparts. Figure 7 shows GUI of our Twitter Trending Topics Classification

[24] http://flash.lakeheadu.ca/~maislam/Mining-Dataset/TestSample/

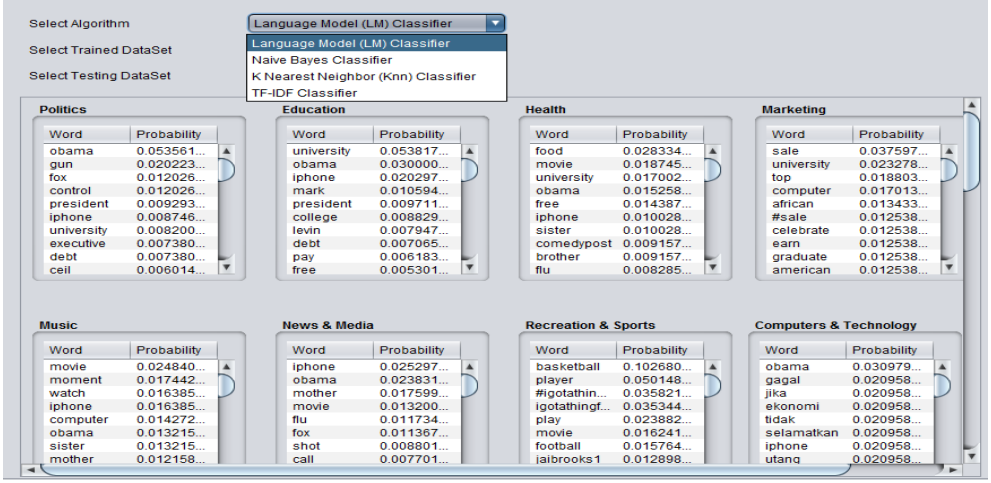


Figure 7. GUI for Twitter Trending Topics Classification

Acknowledgements

Dr. J. Fiaidhi would like to acknowledge the support of NSERC for the research conducted in this article.

References

[1] Y. Hui Lim, "8 Excellent Twitter Analytics Tools to Extract Insights from Twitter Streams", Social Media Today Blog, **(2009)** March 17, http://socialmediatoday.com/index.php?q=SMC/80437.
[2] A. Ritter, S. Clark and O. Etzioni, "Named entity recognition in tweets: an experimental study". Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, **(2011)** July, pp. 1524-1534.
[3] J. Benhardus, "Streaming trend detection in twitter", National Science Foundation REU for Artificial Intelligence, Natural Language Processing and Information Retrieval, University of Colarado, **(2010)**.
[4] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu and M. Demirbas, "Short text classification in twitter to improve information filtering", Proceeding of the 33rd international ACM SIGIR conference on research and development in information retrieval, ACM, **(2010)** July, pp. 841-842.
[5] A. Go, R. Bhayani and L. Huang, "Twitter sentiment classification using distant supervision", CS224N Project Report, Stanford, **(2009)**, pp. 1-12.
[6] S. Kinsella, A. Passant and J. Breslin, "Topic classification in social media using metadata from hyperlinked objects", Advances in Information Retrieval, **(2011)**, pp. 201-206.
[7] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman and J. Sperling, "Twitterstand: news in tweets", Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, **(2009)** November, pp. 42-51.
[8] H. Becker, M. Naaman and L. Gravano, "Beyond trending topics: Real-world event identification on twitter", Proc. of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM'11), **(2011)** July.
[9] A. Zubiaga, D. Spina, V. Fresno and R. Martínez, "Classifying trending topics: a typology of conversation triggers on twitter", Proceedings of the 20th ACM international conference on Information and knowledge management, ACM, **(2011)** October, pp. 2461-2464.
[10] C. Thongsuk, C. Haruechaiyasak and S. Saelee, "Multi-classification of business types on twitter based on topic model", Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference, IEEE, **(2011)** May, pp. 508-511.
[11] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal and A. Choudhary, "Twitter trending topic classification", Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference, IEEE, **(2011)** December, pp. 251-258.
[12] F. Sebastiani, "Machine learning in automated text categorization", ACM computing surveys (CSUR), vol. 34, no. 1, **(2002)**, pp. 1-47.
[13] L. Tenenboim, B. Shapira and P. Shoval, "Ontology-based classification of news in an electronic newspaper", **(2008)**.

[14] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview", International Journal of Data Warehousing and Mining (IJDWM), vol. 3, no. 3, **(2007)**, pp. 1-13.
[15] S. Kiritchenko, S. Matwin and F. Famili, "Functional annotation of genes using hierarchical text categorization", **(2005)**.
[16] D. Koller and M. Sahami, "Hierarchically classifying documents using very few words", **(1997)**.
[17] J. Fiaidhi, S. Mohammed and A. Islam, "Towards Identifying Personalized Twitter Trending Topics using the Twitter Client RSS Feeds", Journal of Emerging Tech. in Web Intelligence, vol. 4, no. 3, **(2012)**, pp. 221-226.

Authors

Jinan Fiaidhi is a Professor of Computer Science and Graduate Coordinators at Lakehead University of Canad. Professional Engineer of Ontario and Adjunct research Professor with University of Western Ontario. Research is on Collaborative Learning, Calm Computing and Machine Learning.

Aminul Islam received his BSc degree in computer science and engineering from Darul Ihsan University, Dhaka, Bangladesh in 2006. Currently he is a Master's student in computer science at Lakehead University, Thunder Bay, Canada.

Sabah Mohammed is a Professor of Computer Science at Lakehead University of Canada. Professional Engineer of Ontario and Adjunct research Professor with University of Western Ontario. Research is on Web Intelligence and Medical Informatics.

Simon Fong is a Professor with the Department of Computer and Information Science at Macau University of China. Research is on Data Analytics, E-Commerce technology, Business Intelligence and Data-mining.

Tai hoon Kim is a Professor of Computer Science, Konkuk University, Korea. Also with GVSA and UTAS, Australia. Vice President of SERSC. Research is on Computer Security.