A Tweet Classification Model Based on Dynamic and Static Component Topic Vectors

Parma Nand, Rivindu Perera, and Gisela Klette

School of Computer and Mathematical Science
Auckland University of Technology
Auckland, New Zealand
{parma.nand,rivindu.perera,gisela.klette}@aut.ac.nz

Abstract. This paper presents an unsupervised architecture for retrieving and ranking conceptually related tweets which can be used in real time. We present a model for ranking tweets with respect to topic relevance in order to improve the accuracy of information extraction.
The proposed architecture uses concept enrichment from a knowledge source in order to expand the concept beyond the search keywords. The enriched concept is used to determine similarity levels between tweets and the given concept followed by a ranking of those tweets based on different similarity values. Tweets above a certain similarity threshold are considered as useful for providing relevant information (this is not part of this paper). We obtained precision values up to 0.81 and F values up to 0.61 for a tweet corpus of 2400 Tweets on the topic related to 2014 NZ general elections.

Keywords: Topic modeling, Natural Language Processing, Text mining, Social Media

1 Introduction

Twitter, as a Social Media Platform (SMP), has been the subject of extensive number of studies for two reasons. Firstly, Twitter is used for posting short public messages rather than group or person to person postings. It is designed for users to post messages with common interests, for example information about politics, events, products, people inter alia. This makes Twitter a unique and rich repository of knowledge on both public, and private aspects of society. The other major reason is that it is easy to retrieve tweets and use them as data for research.

The *Twitter API*[1] allows real time retrieval of tweets based on keywords and other limited criteria such as posting time, tweets from specific accounts and topic popularity. The most common technique to retrieve tweets on a topic is to use a logical combination of keywords or phrases, which will retrieve all tweets containing the keywords. In most cases, this will also download a large amount of

[1] Twitter API: https://dev.twitter.com/

unrelated tweets, since the keywords could have been used in a different context and/or used with a different sense. In order to be able to filter out the irrelevant tweets, there has been substantial efforts (Eg. [1–5])to re-rank the retrieved tweets from the Twitter API according their appropriateness to the topic.

Several AI researchers have exploited the social media attributes of tweets as features to train machine learning algorithms in attempts to rank them according to topic appropriateness. [6] define Twitter Building Blocks (TBBs) as structural blocks in a Twitter message and use these for higher level informational characteristics. As an example, tweets with the same structure as BBC news tweets are likely to be news tweets. The TBBs consist of non-content features such as neighbour TBB type, TBB count, TBB length and OOV (out of vocabulary words). Other works such as [7] and [8] also exploit generic social media as well as Twitter specific features to rank tweets retrieved from the Twitter API using a keyword based query. Some sample features used in these works are the presence or absence of a URL, author authorities, whether the current tweet is a repost(re-tweet), number of re-tweets, hash tag score, whether the current tweet is a reply-tweet and the ratio of OOV words to total number of words. In addition to the social media structural features Duan et. al., also use a content feature. They compute the cosine similarity between each pair of tweets and then use this score in combination with Term Frequency Inverse Document Frequency (TF-IDF) to rank the individual tweets retrieved by the query.

In this paper we present a language model which uses concept enrichment to retrieve and rank tweets by capturing the dynamics of a given topic on a micro-blogging platform (MBP). The model is based on the fact that a trending topic on a MBP consists of two components; a persistent component and a dynamic component. The model uses the information content of the tweets for dynamic component and an external knowledge base, DBpedia, for the persistent component. The model was tested on data from Twitter, however the model can be translated on any MBP since it is based entirely on content, rather than MBP-specific structure related features.

2 Background

Previous works that have dealt with the task of twitter ranking can be categorized into those that make use of a machine learning algorithm and those that use some form of similarity measure. The work in [6] presents a model which uses a fifteen dimension feature vector to train a SVM (Support Vector Machine) model by using a corpus of 2000 human tagged tweets. Each tweet is split into Twitter Building Blocks (TBBs) consisting of the tokens with at least one TBB containing the query term used to search the tweets. The other TBBs contain structural features such as whether the TBB is an URL, whether there are followers of the author and whether the current tweet was retweeted. The paper ([7]) presents a very similar model using 3 sets of features; content relevance, twitter specific features and account authority. This study concludes that account authority and the length of tweets are the best conjunction as features for learning to rank tweets. [8] present another model, however this is completely based on structural features. This study affirms the conclusions from [7], however emphasize that the presence of URL is a stronger feature relevant to ranking.

Unlike above papers, we focus on ranking tweets based entirely on the content of the message rather than other unrelated structural features. The proposed model is similar to the one presented in [?]. In this work the authors use TF-IDF for ranking new tweets based on a background corpus consisting of 150,000 Twitter message corpus. The ranked messages are then merged into topic clusters using Jaccard similarity exceeding 65%. The limitation in O'Connor et. al's model is that the ranking is biased by a static corpus, hence is not completely realtime. [9] present a model which mitigates this limitation by capturing the dynamics of the topics using query expansion. The authors of this work build a background corpus by selecting messages posted closer to the query time using the original query terms. The rationale for this is that messages temporally closer to the query time are more relevant compared to older messages. A weighted mixture of the original query and top n terms from the generated corpus is then used to expand the query to retrieve further messages and rank them. Our model is an extension of the notion of query expansion from [9].

The rationale for query expansion is that tweets about an entity can be expressed by a wide a set of keywords rather than a single or a couple of keywords. When a user wishes to search for tweets pertaining to a topic, he would normally enter either a single or a very small logical combination of keywords resulting in selecting tweets which directly contain the keywords. This would leave out a large proportion of texts which use other keywords relevant to the topic.

A topic on a MBP can be broken down into a persistent and a dynamic component. The dynamic component accounts for the current conversation about an entity and this can rapidly change over time. The persistent component consists of conversation about the more static attributes of the topic. In order to be able to identify a balanced set of tweets one would need to use some combination of the dynamic and the persistent components. We propose a model which uses knowledge infusion from DBpedia to account for the persistent component and the MBP itself for the dynamic component. The information from these sources is combined to form word vectors followed by using a selection of similarity calculators in order to rank the messages. The architecture and the experiments are described in detail in the next section.

3 Model description and Experiment

The proposed query expansion model uses *DBpedia*[2] as the knowledge source, however any other knowledge source such as the *Google* search may be effective as well. DBpedia provides persistent knowledge of about 4.0 million entities, categorized under 529 classes (person, organization, places, etc.). The knowledge is organized as predicates called triples, approximately consisting of a subject,

[2] DBPedia: http://www.dbpedia.org

predicate and object. For example the triple "⟨*New Zealand National Party, leader, John Key*⟩" contains the information about the party leader and similarly "⟨*New Zealand National Party, type, Liberal-conservativeParties*⟩" contains information about the type of party. DBpedia knowledge base is organized into pages corresponding to the pages in Wikipedia, however Wikipedia may contain slightly more information as free text which might not have been structured in the DBpedia knowledge base.

Information was extracted from DBpedia by extracting the predicates Resource Description Framework (RDF) files as described in [10, 11]. The predicates objects and subjects were then extracted from the predicates and sent through a pre-processing module. This module cleaned the noun phrases by removing non English characters, numbers, URLS, punctuations, duplicates and noun phrases which were longer than 50 tokens. The resulting noun phrases was tokenized and the tokens were used to construct the persistent component of the topic vector.

The dynamic component of the word vector was constructed using a set of *seed* tweets. The set of seed tweets is constructed by retrieving the first 100 tweets using only the noun phrase corresponding the topic entity using the Twitter API. The tweets retrieved from the Twitter API was first filtered for locality compatibility. In the case of a location mention in the content of a tweet, we used the location miner from [12] to eliminate tweets which did not belong to the locality of the topic, which was New Zealand for this project, extracted from the query terms. This gives us locality specific tweets that are directly related to the topic entity, however, will also contain other related entities that are typical at the time of the retrieval. The seed tweets were POS tagged using a HMM POS tagger from [13, 14] which is able to identify the syntactical components as well as tweeter specific components such as hash tags URLs, and user mentions.

We downloaded a set of 2400 tweets before the New Zealand general elections at the end of 2014 for a larger research project on the influence of social media on party popularity.

The tweets were download using the keywords pertaining to New Zealand elections such as "*John Key*", "*National Party*" and "*NZ elections*". This resulted in a wide variety of tweets belonging to the wider topic of elections in NZ. The objective for the experiment was to rank the tweets that are relevant to the "National Party" which could be used later for downstream tasks such as sentiment detection. The tweets were manually annotated by a group of 15 post-graduate NLP students as being relevant to National Party or not. The annotators were instructed to take into account the topics relevant directly to National Party as well as the evolving topic temporally relevant to National Party. A selection of 100 tweets were annotated by 4 different annotators with a Cohens Kappa coefficient of 0.87. The annotators classified a total of 591 out of 2400 tweets belonging to the topic of "*National Party in New Zealand*"

The word vector consisting of the persistent and dynamic components were tested with various weights for similarity calculations. We used the following similarity measures: Cosine Similarity, Euclidean Distance, MongeElkan Similarity, Levenshtein Similarity, JaroWinkler Distance, Jaccard Distance and TFIDF

Distance. Similarity algorithms were implemented as described in [15]. Various components of POS components were tested in the similarity computations, the best performance was achieved using combinations of nouns, proper nouns and hash tags, hence all the results reported in this paper are based on tokens corresponding to these three tags. The tagging used in the ranking computations were directly from the POS tagger, hence the ranking results incorporate the propagated errors to all upstream tasks such as tokenization, tagging and chunking.

4 Ranking Results and Discussion

Table 1. Performance Comparison of Similarity Algorithms for top 250 ranked Tweets

Algorithm	**Accuracy**	**Precision**	**Recall**	**F-value**
Cosine	0.85	0.66	0.57	0.61
Jaccard	0.83	0.61	0.53	0.57
Euclid	0.77	0.43	0.37	0.40
Mongee	0.73	0.33	0.28	0.30
Levins	0.71	0.28	0.24	0.26
Jarowr	0.71	0.28	0.24	0.26
TFIDF	0.69	0.21	0.18	0.20

Iinitial experiments were done to determine the best similarity algorithm. We applied 50% weight for the persistent and the same for the dynamic components using a word vector size of 100. The static component words were chosen based on the first 50 tokens from DBpedia triples. The 50 words for the persistent component were chosen from the top, most frequent set comprising of tokens from the noun phrases and hash tags. This topic word vector was then compared with the tweet word vector consisting only of noun phrases and hash tags. Experiments including other components such as @mentions and verb phrases did not yield good results. Table 1 summarizes the results of the similarity computations.

The results show that Cosine similarity was a clear winner with an F-value of 0.61 for selecting the top 250 ranked tweets from the corpus of 2400. Jaccard distance also had a relatively high F-value of 0.57 compared to the rest of the algorithms which had F-values less then 0.4. The rest of the experiments were done using the best performing similarity calculator, Cosine Similarity.

Tweeter ranking was done by using various combinations of topic vectors and various tweet components. The best results were obtained using about half of the topic vector from the persistent topics from DBpedia and the other half from the noun phrases and hash tags from the seed tweets. The comparison vector for the topic specific Tweets was constructed using the most frequent terms from Tweets downloaded using the keywords “John Key”, “National Party” and “NZ elections”.

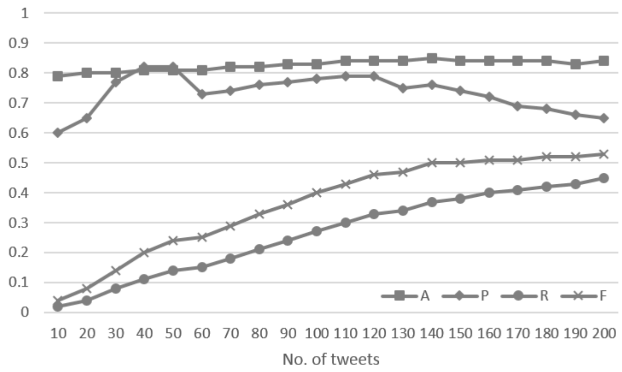

Fig. 1. Results for Retrieving Relevant Tweets with Equal Proportions of Persistent and Dynamic Topics in the Word Vector

The tweets were ranked using amplified cosine similarity values above an arbitrary threshold value of 100. The accuracy, precision, recall and F-values were computed for top tweets ranging from 10, in steps of 10 up a total of 200. The first experiment was done by using a topic vector of only persistent topics consisting of 100 words from the DBpedia page for the New Zealand National Party. The graph in Figure 1 shows results obtained with equal proportion of topics from both persistent and dynamic components. The highest F-value obtained was 0.64, a precision value of 0.81 and a recall value of 0.53 for 190 tweets. The next best result was for persistent only topics with values 0.58, 0.73 and 0.48 respectively.

The results show that external infusion of knowledge for downloading and ranking tweets significantly increases the accuracy. Our proposed language model uses the fact that tweets relevant to a topic would revolve partly around the persistent topic and partly around evolving temporal topics current at the time of retrieval. Model tests show that the best results are achieved when a combination of both persistent topics derived from an external knowledge source and dynamic topic derived from a set of seed tweets are combined.

5 Conclusion and Future Work

We presented a non-learning language model which can be used to retrieve ranked tweets relevant to a topic of interest. The model divides a topic into knowledge around more persistent aspects and those that are transient and temporally relevant. We used DBpedia for the persistent component and a small set of seed tweets for the dynamic component. The dynamic and the persistent combined word vector used with cosine similarity calculator gave an F-value of 0.64 with a precision of 0.81 with a sample size of 2400 tweets. In future we are going to verify the model with more extensive range of topics and with multiple sources of knowledge such as Google Search results.

References

1. Luo, Z., Osborne, M., Petrovic, S., Wang, T.: Improving Twitter Retrieval by Exploiting Structural Information. AAAI (2012)
2. Dong, A., Zhang, R., Kolari, P., Bai, J.: Time is of the essence: improving recency ranking using twitter data. Proceedings of the 19th . . . (2010)
3. Han, Z., Li, X., Yang, M., Qi, H., Li, S., Zhao, T.: Hit at trec 2012 microblog track. Proceedings of Text REtrieval . . . (2012)
4. Efron, M., Golovchinsky, G.: Estimation methods for ranking recent information. . . . and development in Information Retrieval (2011)
5. Luo, Z., Osborne, M., Tang, J., Wang, T.: Who will retweet me?: finding retweeters in Twitter. . . . development in information retrieval (2013)
6. Zhunchen Luo, Miles Osborne, Saša Petrović, T.W.: Improving twitter retrieval by exploiting structural information. AAAI **Proceeding** (2012) 22–26
7. Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.Y.: An empirical study on learning to rank of tweets. Proceedings of the 23rd International Conference on Computational Linguistics (2010) 295–303
8. Nagmoti, R., Teredesai, A., De Cock, M.: Ranking Approaches for Microblog Search. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. Volume 1., IEEE (2010) 153–157
9. Massoudi, K., Tsagkias, M., de Rijke, M., Weerkamp, W.: Incorporating query expansion and quality indicators in searching microblog posts. Proceedings of the 33rd European Conference on Advances in Information Retrieval (2011) 362–367
10. Perera, R., Nand, P.: The Role of Linked Data in Content Selection. In Pham, D.N., Park, S.B., eds.: PRICAI 2014: Trends in Artificial Intelligence. Volume 8862 of Lecture Notes in Computer Science. Springer International Publishing (2014) 573–586
11. Perera, R., Nand, P.: Real Text-CS - Corpus Based Domain Independent Content Selection Model. In: IEEE 26th International Conference on Tools with Artificial Intelligence (ICTAI). (2014) 599–606
12. Nand, P., Perera, R., Sreekumar, A., Lingmin, H.: A Multi-Strategy Approach for Location Mining in Tweets: AUT NLP Group Entry for ALTA-2014 Shared Task. In: Proceedings of the Australasian Language Technology Association Workshop 2014, Brisbane, Australia (2014) 163–170
13. Nand, P., Lal, R., Perera, R.: A HMM POS Tagger for Micro-Blogging Type Texts. In: Proceedings of the 13th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2014). (2014)
14. Nand, P., Perera, R.: An evaluation of pos tagging for tweets using hmm modeling. 38th Australasian Computer Science Conference (2015)
15. Chapman, S.: Simmetrics. URL http://sourceforge. net/projects/simmetrics/. SimMetrics is a Similarity Metric Library, eg from edit distances (Levenshtein, Gotoh, Jaro etc) to other metrics,(eg Soundex, Chapman). Work provided by UK Sheffield University funded by (AKT) an IRC sponsored by EPSRC, grant number GR N **15764** (2009)