Entity Tracking in Real-Time using Sub-Topic Detection on Twitter

Sandeep Panem, Romil Bansal, Manish Gupta, Vasudeva Varma

International Institute of Information Technology, Hyderabad, India

Abstract. The velocity, volume and variety with which Twitter generates text is increasing exponentially. It is critical to determine latent sub-topics from such tweet data at any given point of time for providing better topic-wise search results relevant to users' informational needs. The two main challenges in mining sub-topics from tweets in real-time are (1) understanding the semantic and the conceptual representation of the tweets, and (2) the ability to determine when a new sub-topic (or cluster) appears in the tweet stream. We address these challenges by proposing two unsupervised clustering approaches. In the first approach, we generate a semantic space representation for each tweet by keyword expansion and keyphrase identification. In the second approach, we transform each tweet into a conceptual space that represents the latent concepts of the tweet. We empirically show that the proposed methods outperform the state-of-the-art methods.

Key words: Sub-Topic Detection, Clustering, Entity Tracking, Text Mining

1 Introduction

In the recent past, Twitter has been widely used for spreading the social pulse about real world entities. Mining sub-topics from entities helps in trend analysis, social monitoring, topic tracking and reputation mining. "Topics" on Twitter relate to major events in the real world; "sub-topics" on the other hand are fine-grained aspects of such events. For example, consider the tweet, "Recently listed on MLS: 2003 Volvo VHD64B200 #mixer from Transport Truck Sales in Kansas City, KS". Here the sub-topic is "buying or selling of trucks" and the topic is "Volvo". Existing topic detection methodologies are generally based on probabilistic language models, such as Probabilistic Latent Semantic Analysis (PLSA) [7] and Latent Dirichlet Allocation (LDA) [4]. By exploiting tweet contents, LIA at CLEF 2013 [2] applied a large variety of machine learning methods for clustering of tweets. However, these methods require the number of topics as an input, and assume that a single document contains rich information, which is not applicable to microblogs. REINA at CLEF 2013 [2] used similarity matrix and community detection techniques for topic detection. UNED_ORM at CLEF 2013 [1] experimented with approaches like agglomerative clustering based on term co-occurrences and clustering of wikified concepts. Few limitations of these approaches are (1) they often mix multiple incoherent sub-topics together, and (2) they cannot find novel topics in streaming scenarios as they need all the data at once. Our two phase clustering approach as described in Section 2 deals with the above mentioned issues by mining sub-topics from streaming text. The proposed clustering is based on a novel representation of tweets using

various combinations of concepts, keyphrases and keywords with appropriate weights assigned to them. We improve the accuracy of detecting sub-topics by discarding less frequent concepts. Also, *batched* updates ensure that our system is efficient enough to be practical.

2 Approach

In this paper, we propose the following two approaches to tackle the problem of dynamic clustering by exploiting the semantic structure of tweets.

1. Semantic Space Representation (*SSR*) based approach
2. Concept Space Representation (*CSR*) based approach

Both the approaches consist of two phases: an offline cluster generation phase and an online cluster maintenance phase. In the offline phase, a graph-based clustering algorithm is used to obtain the initial clusters from a few tweets. These initial clusters are later used in the online phase to cluster new tweets from the tweet stream and also to update the clusters themselves. Figures 1 and 2 illustrate the offline and online phases for the two approaches respectively.

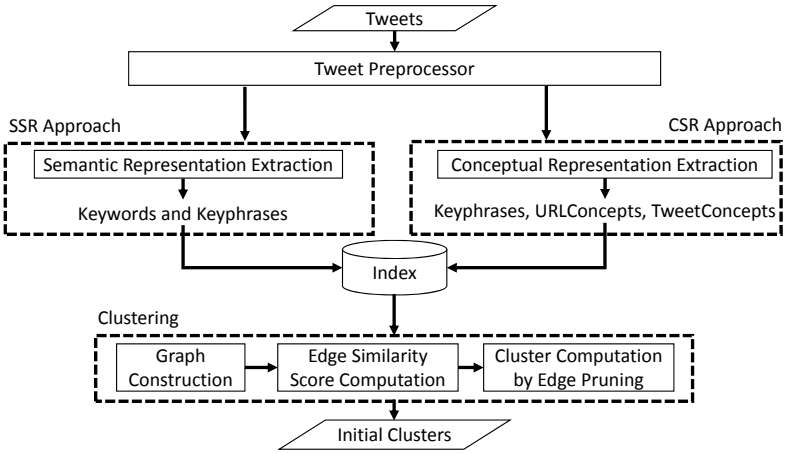

Fig. 1. Offline Initial Cluster Generation Phase for Both Approaches

Semantic Space Representation (*SSR*) based Approach
In the offline phase, we first preprocess the tweets by removing stopwords and performing POS tagging and URL extraction. We consider only English tweets across multiple domains maintaining the heterogeneity. We extract the longest sequence of nouns as well as proper nouns and keyphrases using POS chunking [5]. We consider nouns, hashtags and proper nouns as keywords and enhance them by finding the synonyms using WordNet and extracting top n words from the page content of the URL contained in

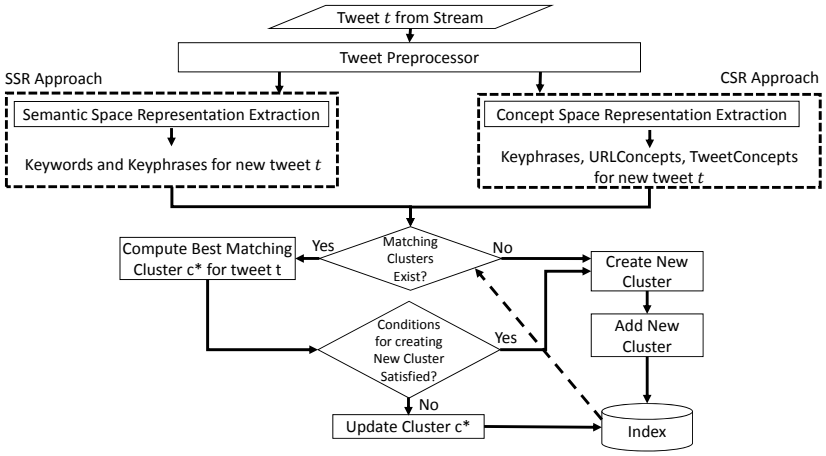

Fig. 2. Online Cluster Maintenance Phase for Both Approaches

the tweet. Thus, the semantic representation of a tweet t consists of keywords ($KW(t)$) and keyphrases ($KP(t)$). Next, we build a graph with tweets as nodes and the similarity between two tweets representing the edge weight. The similarity between two tweets, t_1 and t_2 is computed as shown in Eq. 1.

$$sim(t_1, t_2) = w \times sim(KP(t_1), KP(t_2)) + (1 - w) \times sim(KW(t_1), KW(t_2)) \quad (1)$$

where,

$$sim(KP(t_1), KP(t_2)) = |KP(t_1) \cap KP(t_2)| \quad (2)$$

and

$$sim(KW(t_1), KW(t_2)) = |KW(t_1) \cap KW(t_2)| \quad (3)$$

Here, w denotes the weight given to the keyphrases and $(1 - w)$ denotes the weight given to the keywords. For our experiments we set $w = 0.6$. We rank the edges based on the similarity among tweets and prune the ranked edges by removing low weight edges until all the vertices continue to be covered by the remaining edges. We then cluster the tweets based on nearest-neighbors similarity. Each cluster is stored along with its keywords and keyphrases in the index.

In the online phase, we process each new tweet t by first extracting its keywords and keyphrases. We then query the index for clusters containing the tweet's keywords and retrieve the cluster c^* with the highest score. Score of each cluster is based on the similarity between the tweet and the cluster in terms of their keywords and keyphrases. Tweet t is assigned to the cluster c^* or to a new cluster based on the following intuitive rules. If no clusters in the index match with the tweet keywords, tweet is assigned to a new cluster. Sometimes even though there is a match in the set of keywords, the tweet may belong to a very different sub-topic compared to the sub-topic of the most similar cluster c^*. To avoid incorrect cluster assignment for the new tweet, we used

"Wikipedia title matching" to distinguish sub-topics related to a particular topic. We compare tweet's keyphrases with Wikipedia titles by performing a substring match. If at least one keyphrase occurs in one of the titles of a Wikipedia page and if none of the tweet's keyphrases match with the keyphrases of the matched cluster c^*, then we create a new cluster. Otherwise we assign the tweet to the cluster c^*. For example, consider the tweet "The Volvo ocean race has started this year on a high range in England". Based on the simple keyword match, this tweet would get assigned to the cluster containing "Volvo" and "England", but in reality it should form a new cluster namely "Volvo ocean race".

Concept Space Representation (*CSR*) based Approach

A lexical mismatch is caused by occurrence of different words in the tweet which are otherwise semantically related. Transformation of tweets into concept space can help reduce the lexical mismatch. This can enable matching even those tweets that are semantically relevant to each other but do not have any overlapping words. We obtain the semantic representation of a tweet by extracting concepts for a tweet using the TagMe [6] API. The API takes short text snippets as input, disambiguates the entities in the text, and maps these entities to Wikipedia pages. Using this API, we represent each tweet t conceptually as a combination of keyphrases ($KP(t)$), URLConcepts ($UC(t)$) and TweetConcepts ($TC(t)$). The offline phase for the *CSR* based approach is similar to the one for the *SSR* based approach.

In the online phase, we query the index for clusters containing the tweet's keywords and retrieve the k nearest clusters. Similarly, we retrieve the top k nearest clusters for URLConcepts and TweetConcepts. For our experiments, we set k as 10. We then assign a score $R(c)$ to cluster c as shown in Eq. 4.

$$R(c) = \frac{\alpha}{R_{KP}(c)} + \frac{\beta}{R_{UC}(c)} + \frac{\gamma}{R_{TC}(c)} \tag{4}$$

Here, α, β and γ are the weights given to keyphrases, URLConcepts and TweetConcepts respectively such that $\alpha + \beta + \gamma = 1$. Here $R_{KP}(c)$, $R_{UC}(c)$, $R_{TC}(c)$ are the ranks of the cluster retrieved when queried with keyphrases, URLConcepts and TweetConcepts respectively. A new tweet t is either assigned to the cluster c^* with the highest score or to a new cluster. If no clusters in the index match with the tweet keywords, the tweet is assigned to a new cluster. The tweet is assigned to the cluster c^* if at least one keyphrase matches between cluster c^* and tweet t, or at least one concept matches other than the entity name. Otherwise the tweet t is assigned to a new cluster.

Maintaining Cluster Purity and Cluster Labels

We preserve the purity of clusters, by removing the irrelevant concepts from the clusters at regular intervals of tweet arrivals into the clusters. After every m (we set m to four) consecutive tweets a cluster receives, we update the cluster by storing only the most frequently occurring concepts. The concepts of tweets in the cluster, and their frequency values are updated. In *SSR*, we label the clusters using the top occurring keyphrase and keyword. In CSR, we define cluster label using the top occurring keyphrase and concept.

As the labels of the cluster change frequently with the incoming tweets, we update the labels of the clusters at regular intervals, δt. For our experiments, we set δt to fifty tweets.

3 Experiments

We focus on the task of entity tracking using sub-topic detection. For the offline phase, tweets can be collected by querying for the entity name on Twitter. If there are no initial tweets related to an entity then the online phase starts with an empty cluster set. For our evaluation, we use RepLab (CLEF) 2013 [2] dataset. The dataset contains tweets for 61 entities. Each entity has about 700 tweets for training and 1500 tweets for testing. In the offline phase, we use the training tweets to obtain seed clusters, which are then used in the online phase to cluster test tweets. The data sets are manually labeled by expert annotators.

For evaluation we use two complementary measures, Reliability and Sensitivity as defined by Amigó et al. [3]. Let us consider a system output X and a gold standard $\mathcal{G}$, which are both a set of document relationships $r(d, d')$. The Reliability (R) of relationships in the system output is the probability of finding them in the gold standard. The Sensitivity (S) of predicted relationships is the probability of finding them in the system output when they appear in the gold standard.

Table 1 shows the system performance with various values of α, β, and γ. The setting $\alpha = 0.3, \beta = 0.2$ and $\gamma = 0.5$ gave the best results. We infer that both the keyphrases and TweetConcepts features are equally important. Because many tweets do not contain a URL, low weight is assigned to the URLConcepts. Table 2 compares the performance of various methods. Here *Baseline* is the memory-based learning baseline supplied by RepLab. We compare our results with *UNED_ORM*, *REINA*, *LIA* (teams that participated in RepLab 2013 [2]) whose approaches are described in Section 1. The total number of sub-topics in the dataset were 9570. The proposed *SSR* and *CSR* based approaches detected 7545 and 8633 sub-topics respectively. We observe that for a few tweets, the concepts retrieved from TagMe were relatively inaccurate, and this resulted in lower reliability for *CSR*. *SSR* has higher reliability compared to *CSR* because most of the relationships predicted by the system are also found in gold standard. However, the sensitivity is lower for *SSR* because the number of discovered relationships in the system are less, as excessive keyword matching (probably because of WordNet usage) caused some sub-topics to merge into a single sub-topic. The higher sensitivity of *CSR* as compared to that of *SSR* is because *CSR* has higher coverage of relationships over the gold standard.

CSR based approach maintains the consistency in both, identifying the number of sub-topics as well as their presence in the gold dataset for each entity. The $F1$ measure values are calculated for each topic individually and averaged over all the topics. As shown in Table 2, the proposed approach achieves the highest $F1$ measure value. The increase in the $F1$ measure value by **16.9%** as compared to the *Baseline* and $\sim$**1%** as compared to the best system in RepLab 2013 indicates that the proposed *CSR* approach performs better than the state-of-the-art methods.

Table 1. Accuracy Results for Various Values of α, β and γ

α	β	γ	R	S	$F1$	#Sub-Topics
0.5	0.2	0.3	0.303	0.521	0.338	8586
0.3	**0.2**	**0.5**	**0.304**	**0.516**	**0.339**	**8633**
0.4	0.3	0.3	0.304	0.522	0.338	8794
0.6	0.2	0.2	0.303	0.512	0.335	8785
0.2	0.3	0.5	0.305	0.516	0.337	8760
0.2	0.4	0.4	0.303	0.529	0.338	8685

Table 2. Performance Comparison of Various Methods

Method	R	S	F
CSR	0.304	**0.516**	**0.339**
UNED_ORM	0.460	0.320	0.330
REINA	0.320	0.430	0.290
SSR	**0.496**	0.203	**0.259**
LIA	0.220	0.350	0.250
Baseline	0.150	0.220	0.170

4 Conclusions

In this paper, we have explored a novel approach by exploiting the semantic and concept based representations of tweets for sub-topic clustering. In the *SSR* based approach, we used keywords (WordNet synonyms, URL keywords), keyphrases and Wikipedia title matching (as criterion for creating new cluster) as features. To handle the issue of over-matching of keywords due to WordNet usage, we propose the *CSR* based approach. In the *CSR* based approach, we used TweetConcepts, URLConcepts and keyphrases as features. We maintain the purity of clusters by periodically cleaning up the clusters. Experiments on the RepLab (at CLEF 2013) dataset showed that the proposed approach achieves significant performance gains over the baseline and other systems using metrics like Reliability, Sensitivity and $F1$ measure. In the future, we would like to extend this study by incorporating the similarity between concepts in the *CSR* based approach.

References

1. E. Amigó, A. Corujo, J. Gonzalo, E. Meij, and M. de Rijke. Overview of RepLab 2012: Evaluating Online Reputation Management Systems. In *Proc. of the* 3^{rd} *Intl. Conf. of the CLEF Initiative*, 2012.
2. E. Amigó, J. C. de Albornoz, I. Chugur, A. Corujo, J. Gonzalo, T. Martín-Wanton, E. Meij, M. de Rijke, and D. Spina. Overview of RepLab 2013: Evaluating Online Reputation Monitoring Systems. In *Proc. of the* 4^{th} *Intl. Conf. of the CLEF Initiative*, pages 333–352, 2013.
3. E. Amigó, J. Gonzalo, and F. Verdejo. A General Evaluation Measure for Document Organization Tasks. In *Proc. of the* 36^{th} *Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 643–652, 2013.
4. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, Mar 2003.
5. K. S. Dave and V. Varma. Pattern Based Keyword Extraction for Contextual Advertising. In *Proc. of the* 19^{th} *ACM Intl. Conf. on Information and Knowledge Management (CIKM)*, pages 1885–1888, 2010.
6. P. Ferragina and U. Scaiella. TagMe: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *Proceedings of the* 19^{th} *ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1625–1628. ACM, 2010.
7. T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proc. of the* 22^{nd} *Annual Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 50–57, 1999.