Dynamic Theme Tracking in Twitter

Liang Zhao  
Virginia Tech  
liaoz8@vt.edu

Feng Chen  
University of Albany, SUNY  
fchen5@albany.edu

Chang-Tien Lu and Naren Ramakrishnan  
Virginia Tech  
ctlu@vt.edu, naren@cs.vt.edu

I. SCORE FUNCTION IN EQUATION 6

\[
s^{(t)}(W_j|D_k) = \begin{cases} 
  s^{(t)}_0(W_j|D_k), & f^{(t)}(W_j,D_k) \neq 0 \\
  0, & f^{(t)}(W_j,D_k) = 0 
\end{cases},
\]

where \( f^{(t)}(W_j,D_k) \) is the frequency of term \( q_t \) in tweet \( D_k \) and 
\[
s^{(t)}_0(W_j|D_k) = \log p_{ML}^{(t)}(W_j|D_k) - \log p_{ML}(W_j|C_t) 
= \log f^{(t)}(W_j,D_k) - \log \sum_{W_i \in D_k} f^{(t)}(W_i,D_k) - \log \frac{1}{|D|} 
\]

where \( c^{(t)}(W_j,D_k) \) is a boolean value such that \( c^{(t)}(W_j,D_k) = 1 \) means the term \( W_j \) appears in the tweet \( D_k \) while \( c^{(t)}(W_j,D_k) = 0 \), otherwise. The notation \( W_j \in D_k \) signifies that the term \( W_j \) is contained in the tweet \( D_k \).

II. TIME COMPLEXITY ANALYSIS

The time consumption consists of two parts: 1) calculation of the term dependencies, as shown in Steps 2-6 in Algorithm ??; and 2) parameter optimization, as shown in Steps 8-15.

By exploiting conditional independence in the chain, the time complexity of the term dependencies calculation is successfully reduced from exponential to linear in the number of variables being marginalized [1]. Considering this, the total time complexity of the first part is \( O(\Phi \cdot |D|) \), where \( \Phi = \bar{n}_u + \bar{n}_a + \bar{n}_r + \bar{n}_f + |q| \), in which \( \bar{n}_u, \bar{n}_a, \bar{n}_r, \bar{n}_f, |q| \ll |D| \) denotes the average number of the terms in a tweet, the average number of the tweets that have a replying relationship with a random tweet, the average number of posts by the user, the average number of friends of a user, and the number of theme query terms, respectively.

The time complexity of the second part is \( O(L(R_1 + R_2)) \), where \( L \) is the number of iterations in the parameters estimation of DQE. \( R_1 \) and \( R_2 \) are the time consumptions of Newton’s method for solving the problems in Equations ?? and ??, respectively. In total, the total time complexity is: \( O(\Phi \cdot |D| + L \cdot (R_1 + R_2)) \).

III. COMPARISON METHODS

The proposed DQE is compared with 5 existing methods and 3 baselines for this study:

Supervised topic models (STM) [3]: STM infers topics appropriate for use in a given classification problem. In the classified theme-related tweets for each date, the theme snapshot is calculated as the proportion of the terms’ frequencies. The number of topics is set to 10, as in the original paper. 11,533 manually labeled tweets, of which there are 5,386 positive (theme-related) and 6,147 negative (not theme-related) are used as the training set. All the terms they contain are used as features.

Query Expansion (QE) [6]: For each date, this method generates expanded query to retrieve theme-related tweets, where the proportion of term frequencies designates as the theme snapshot. Following the original paper, the top 10 expanded terms were selected, and tweets containing at least one of them are retrieved.

Dynamic topic models (DTM) [2]: DTM models the dynamics of topic evolution by aligning the latent topics via a Kalman filter. The number of latent topics used here is set as 100. The topic with the best-performance is used as the theme snapshot for this comparison.

TEDAS [4]: For each date this method performs a query expansion to retrieve tweets and uses a classifier to refine them, where the proportion of term frequencies designates as the theme snapshot. The classifier’s feature set consists of both text features (i.e., terms in the tweet corpus) and social network features (i.e., hashtags, url, and mentioned user). The training set is the same as that used for STM.

Language model-based (LMB) approach [5]: For each date, this method utilizes a language model with “normalized stupid smoothing” [5] to extract theme-related tweets, where the proportion of term frequencies designates as the theme snapshot. The training set is the same as that used for STM.

Earthquake detection (ED) [7]: For each date, ED utilizes a classifier to retrieve tweets, where the proportion of term frequencies designates as the theme snapshot. The training set here is once again the same as that for STM. The vocabulary of all the terms in the training set is the feature set.

DQE-C, DQE-R, and DQE-A: These are the 3 baselines of the proposed DQE and are the same as DQE except for the calculation of term dependencies such that: 1) “DQE-C” only considers the “co-occurrence” relationship; 2) “DQE-R” considers both the “co-occurrence” and “relying” relationships; and 3) “DQE-A” considers “co-occurrence”,
“replying”, and “authorship”. DQE-C, DQE-R, and DQE-A are trained the same way as DQE. The initialization of the theme queries for the baselines is the same as that for DQE, as described in Section ??.

REFERENCES