Constructing Knowledge Base's Category Topics for Transfer Learning in Accurate Microblog Event Detection

Abstract

Many web applications need the accurate event detection technique on microblog stream. But the accuracy of existing methods are still challenged by microblog's short length and high noise. We develop a novel lightweight transfer learning method LTDETECTOR to deal with the task. LTDETECTOR bases on two facts, that microblog is short but can be enriched by knowledge base semantically with transfer learning; and to make the transfer efficiently and effectively, the scale of knowledge should shrink without degrading. The following contributions are made in LTDETECTOR. (1) We formalize the knowledge base's hierarchical structure into three parts including taxonomy graph, category-page bipartite graph, and page-content map; and further shrink the scale of knowledge to category-word level by utilizing the structure. Specifically, category's highly related words and their chi-square scores, namely category's *KB State*, are condensed. (2) Category's related words in microblog stream, namely category's *Stream State*, are learned from our KB-Prior-LDA, which exploits the word co-occurrences and transfers the knowledge base's category topics into microblogs. (3) Events are further detected on meaningful categories' *Stream States*, which filter out the noise accurately. (4) Experiments on the benchmark *Edinburgh twitter corpus (30 million tweets)* validate the effectiveness of our proposed transfer learning method. LTDETECTOR achieves high accuracy, promoting the precision by 9% without sacrificing the recall rate.

1 Introduction

Many web applications need the accurate event detection technique on microblog stream, such as public opinion analysis(Thelwall *et al.* 2011), public security(Li *et al.* 2012b), and disaster response(Yin *et al.* 2012), etc. Although event detection has been a research topic for a long while(Allan *et al.* 1998), event detection in microblog stream is still challenging(Atefeh and Khreich 2015). According to (Huang *et al.* 2016), the characteristics of microblog, which is fast changing, high noise, and short length, raise the challenge.

Knowledge base can be a good supplementary for event detection on microblog stream. Different from the not-well-organized microblog stream, knowledge base (e.g. Wikipedia) is constructed elaborately and contains rich information. For example, the microblog message "Possible Ft. Hood Attack Thwarted (2011-07-28)" is short, but still comprehensible because the words "*Ft. Hood*" is included in the wiki page "*Ft. Hood*", and belongs to the category "*Military*". By reading these two wiki pages, the model easily understands the example tweet is talking about something related to *Military*. In other words, knowledge base enriches the linkages between words, and provides more comprehensive context for microblogs. Since the transfer learning(Pan and Yang 2010) aims at utilizing the extra information restored in the source dataset to benefit the target dataset, it provides a feasible way to enhance the event detection in microblog stream.

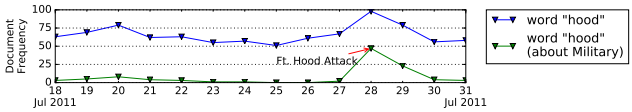

Figure 1: Hood[1][2]

But it's non-trivial to conduct transfer learning from knowledge base to microblog stream directly. The existing RDF model(Klyne and Carroll 2006) lacks an efficient quick mechanism to transfer the knowledge, since it's mainly designed for managing knowledge as tuples on graph. And the query on large graph is also very expensive(Huang *et al.* 2011), which is not suitable for the scene of quickly and acuaretely detecting events. What most meet the efficiency demand is Twevent(Li *et al.* 2012a), but it's limited to filtering out meaningless microblogs by looking up Wikipedia. Some event, which may contains the segments not restored in Wikipedia, may be dropped incorrectly.

In our paper, to balance the performance and the cost of transfer learning, new data structure *KB states* is proposed. It bases on the following facts. The knowledge base's three fold structure includes the taxonomy graph (*class* $\rightarrow$ *subclass*), the category-page bipartite graph (*class* $\rightarrow$ *instance*), and the page-content map (*instance* $\rightarrow$ *content*) relations in the knowledge base. In terms of concept level, the latter part is finer than the former. And the last page-content map goes into the detail at the word level. By considering these three parts together, we can extract a class or category's highly related words (evaluated by chi-square score in Section 3.1), and restore them in the newly proposed data structure *history*

states. In transfer learning, *history states* play an important role at speeding up building the linkage between the word in microblog and the high level concept in knowledge base.

More than that, we propose a new quick and lightweight probabilistic model KB-Prior-LDA for transfer learning. After deriving KB-Prior-LDA (detailed in Section 3.2), it reveals that whether a word in microblog links to a high-level category in knowledge base depends on three factors, the word's score in the category's *KB state*, the linkages of microblogs' other words, the linkages of the word types in the same time window. All these statistics are easy to gain, which leads the model efficient. What KB-Prior-LDA learned is summarized as *recent states*, shown in Figure 2. Since the recent states have filtered out meaningless words, the detected events on recent states is much more accurate than other methods which don't do the transfer learning. KB-PRIOR-LDA, *history states*, and *recent states* forms up our proposed event detection method LTDETECTOR. Our experiment on the real dataset *Edinburgh twitter corpus (30 million tweets)* validates the effectiveness of our proposed method LTDETECTOR.

The transfer learning based event detection method LTDETECTOR also coincides with the news value theory(Caple and Bednarek 2013), which emphasizes the *news unexpectedness* – "the unpredictable or the rare is more newsworthy than the routine"(Bell 1991). Following this guideline, if the journalist knew the history states, aka, the "routine things" of everyday better, he/she would tend to choose more unexpected news to report.

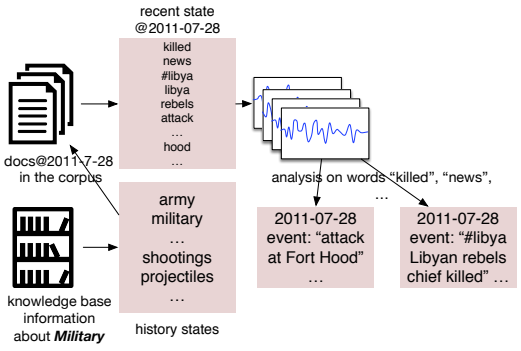

Figure 2: LTDETECTOR's process flow, taking *Military* related events in cyberspace as an example. LTDETECTOR initializes the normal states about military from knowledge base, learns the topical words further within the time windows on the target corpus, detects events.

2 Related Works

Event Detection. There are two kinds of event detection methods based on how it utilize the history data, unsupervised and supervised learning.

The unsupervised methods, including UMass(Allan *et al.* 2000) and BurstyBTM(Yan *et al.* 2015b), use *the data within the recent time windows* to help to decide whether the incoming article is related to a new event. This kind methods can be implemented by clustering of articles(Allan *et al.* 2000; Petrovic *et al.* 2010b; Wurzer *et al.* 2015), word frequencies(Mathioudakis and Koudas 2010)(Weng *et al.* 2011), or topic modelling(Diao *et al.* 2012)(Yan *et al.* 2015b) in the recent time windows. Taking the clustering of articles as an example, they model the occurred events as clusters, and link the incoming article with an already existed event cluster or assign it as the new detected event. The decision is based on whether the dissimilarity between the incoming article and existed event clusters is over the user-specified threshold.

The reason why this kind methods cannot achieve high precision and high recall simultaneously lies in that they lack the accumulation of history knowledge. This kind methods "understand" the document only by the recent data and the frequencies of the data, but the "understanding" is not so confident that they still need the specific threshold to help to make decision. In the journalist metaphor of news value theory, they have the limited knowledge of normal states (aka "routine things") due to the limited history information used. On the one hand, UMass(Allan *et al.* 2000) prefers the lower threshold to enlarge the recall but suffers the precision. On the other hand, BurstyBTM needs the appropriate number of topics to run the topic modelling, and detects the "large" events that associate with many articles but ignores the "small" ones.

The supervised thoroughly relies on the *history data*. The typical implementation is to train the event triggers(Li *et al.* 2013)(Nguyen and Grishman 2015) from history labeled data, then apply the trained model to detect the event related articles, which contain the triggers such as the word *attacked* for the *public security* events. This kind methods suffer from the loss of recall. There are two reasons. (1) The model does not utilize the history data sufficiently. Usually, the event trigger based method mainly focus on the event related verbs, but often ignores the other word types. (2) The model lacks the update mechanism. It only restores the history knowledge but without updating. Some new event related articles may be mistakenly filtered out because not containing previous trained triggers. For example, the tweet "Russia's intervention in Syria began in September 2015" without an event trigger, which is really related to a *public security* event, may be omitted by the system.

Knowledge Base The existing RDF model(Klyne and Carroll 2006) lacks an efficient quick mechanism to transfer the knowledge, since it's mainly designed for managing knowledge as tuples on graph. And the query on large graph is also very expensive(Huang *et al.* 2011), which is not suitable for the scene of quickly and accurately detecting events.

3 Proposed Method

Knowledge base often contains much information about what happened. But the existing RDF model(Klyne and Carroll 2006) on knowledge base lacks an efficient mechanism to transfer the information from knowledge base to text stream. With the help of knowledge base's structure (detailed in sec 3.1), we provide a lightweight model to transfer the information in knowledge base to text stream.

History state is the proposed data structure to restore the information about what happened. Initially, *history state* extracts category-related history data into a set of tuples, which weighs the importance of given words to the specific category, defined in Definition 1 formally. Taking the *military* category in Figure 2 as an example, the history state of *military* contain the words *army*, *military*, and *shootings* etc.

Recent state is the proposed data structure to maintain the information about what's happening. It extracts category-related words from incoming text stream, for each time window. Still taking the *military* category in Figure 2 as an example, based on extracted history states, our proposed probabilistic continuous learning model HS-PRIOR-LDA can recognize the words *libyan* and *rebel* in the time window 2011-07-28 in the text stream are related to the category *military*. Furthermore the time series analysis on the neighbouring recent states can recognize the set of events' candidate words, and finally detects the events in the text stream. In the example of Figure 2, the words *libyan* and *rebel* are related to the *military* event "#libya Libyan rebels chief is killed (2011-07-28)".

Definition 1 (History State) *The history state of the specific category is defined by a set of tuples, in which the first element is the word $w_i^{(c)}$ related to the category c, and the second element is the chi-square score $chi(c, w_i^{(c)})$ under the category c. And we denote the history states of category c as $\boldsymbol{h}_c = \{< w_i^{(c)}, chi(c, w_i^{(c)}) >\}_{i=1,\ldots,N_c}$.*

Definition 2 (Recent State) *The recent state at time t of the specific category c is defined by a set of tuples, and denoted as $\boldsymbol{r}_{c,t} = \{< w_i^{(c)}, n(c,t,w_i^{(c)}) >\}_{i=1,\ldots,N_c}$, in which word $w_i^{(c)}$ is related to the category c, and $n(c,t,w_i^{(c)})$ is its document frequency in the time window t.*

Section 3.1 explains the details of how to initialize history states from knowledge base. Section 3.2 shows how to perform transfer learning for recent states. And section 3.3 interprets the high accuracy event detection based on the processed recent states.

3.1 History States' Extraction

In this part, we discuss in detail how to initialize the history states from the given knowledge base. The knowledge base such as Wikipedia has the structure of classes, subclasses, instances, and the edges between them. This structure usually can be represented as triples in RDF graph(Klyne and Carroll 2006), which is adopted to build DBPedia(Auer *et al.* 2007) and YAGO(Suchanek *et al.* 2007) from Wikipedia. But the reasoning and maintenance of knowledge on the graph is usually expensive(Broekstra and Kampman 2003; Bursztyn *et al.* 2015). To make a trade-off between cost and performance, we use the lightweight data structure *history state* to represent the knowledge about what happened, which is extracted from the knowledge base. And the knowledge base's threefold structure $G^{(0)}$, $G^{(1)}$, $G^{(2)}$ benefits the extraction of history states.

Taxonomy Graph $G^{(0)}$. The directed edges in $G^{(0)}$ represent the *class→subclass* relations in the knowledge base. Taking Wikipedia for example (Figure 3), the node *Main topic classifications*[3] has the subclass *Society*, further contains the subclass *Politics*, which is the ancestor of the subclass *Military*. As $G^{(0)}$ is not a Directed Acyclic Graph originally(Faralli *et al.* 2015), we remove the cycles according to nodes' PageRank-HITS(Yan *et al.* 2015a) score. Specifically, the edges *class→subclass* are preserved only when the node *class* has the higher PageRank-HITS score than the node *subclass*, which is shown in the line 2 of Algorithm 1. After removing cycles, the taxonomy structure on the knowledge base is better represented by the directed acyclic graph $G^{(0)'}$. As shown in line 3 of Algorithm 1, by visiting the category *Military* in the DAG $G^{(0)'}$, the breadth-first traverse can reach its successor sub-category nodes such as *Firearms*, *The World Wars*, and *World War II*, etc.

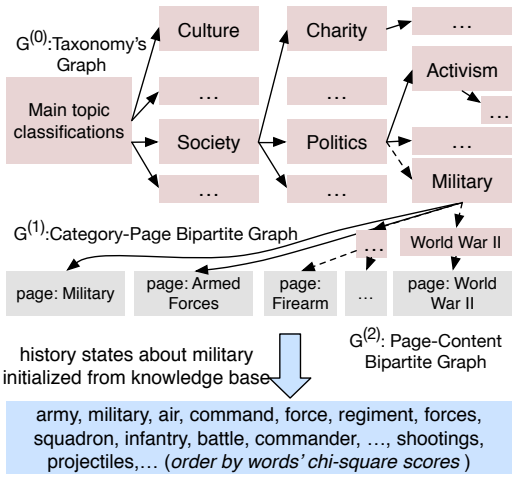

Figure 3: Illustration on how to initialize history states from Wikipedia, taking *Military* as an example.

Category-Page Bipartite Graph $G^{(1)}$. The directed edges in $G^{(1)}$ represent the *class→instance* relations in the knowledge base. In Wikipedia, by considering $G^{(0)'}$ and $G^{(1)}$ together as shown in line 5 of Algorithm 1, we can get all the pages related to the given category.

Page-Content Map $G^{(2)}$. For a specific Wikipedia dumps version, the edges *page→content* in $G^{(2)}$ define a one-to-one mapping. There are a bulk of history information restored in the wiki text content. In order to extract the key words in wiki page's content, which distinguish it from other pages, we use the chi-square statistics(Yang and Pedersen 1997)(Liu *et al.* 2009) to measure the importance of each word for the specific page.

For a given category in $G^{(0)'}$, chi-square statistics also can evaluate the importance of each word appeared in text contents. For example, the chi-square statistic of the term *shooting* under the category *military* is 2888.7 under chi-square test with 1 degree of freedom. That means the term *shooting* is highly related to the category *military*. Finally,

[3] https://en.wikipedia.org/wiki/Category:Main_topic_classifications

we can get the category's history state, which are composed by the category related words and their corresponding importance.

Algorithm 1: History State Initialization from Knowledge Base

Input: Taxonomy's Graph $G^{(0)}$, Category-Page Bipartite Graph $G^{(1)}$, Page-Content Bipartite Graph $G^{(2)}$, topic related category node c
Output: History state $\boldsymbol{h}_c$ on category c

$Pages(c) \leftarrow \varnothing, \boldsymbol{h}_c \leftarrow \varnothing$
DAG ${G^{(0)}}' \leftarrow$ Remove Cycles of $G^{(0)}$ by nodes' HITS-PageRank scores.
$SuccessorNodes(c) \leftarrow$ Breadth-first-traverse(${G^{(0)}}', c$)
for $node \in SuccessorNodes(c)$ **do**
    $Pages(c) \leftarrow Pages(c) \cup G^{(1)}.neighbours(node)$
Word frequency table $n(c, .) \leftarrow$ do word count on the text contents of $Pages(c)$
Word frequency table $n(All, .) \leftarrow$ do word count on the text contents of all pages in $G^{(2)}$.
for *word w in WordFrequencyTable(All).keys()* **do**
    $chi(c, w) \leftarrow w$'s chi-square statistics on $WordFrequencyTable(c)$ and $WordFrequencyTable(All)$.
    $h_{c,w} \leftarrow chi(c, w)$
return $\boldsymbol{h}_c$

Considering the full Wikipedia's contents, history state evaluates the importance of word to the concerned category accurately. For example, in *Military*'s history state, the top words are *army*, *military*, *air*, *command*, *force*, and *regiment*, etc. The document which contains these words is related to the category *Military* with high probability. We further discuss how to apply history states to the text stream on the following subsection 3.2.

3.2 Recent States' Maintenance

In this subsection, we describe how the proposed probabilistic model HS-PRIOR-LDA utilizes the history states to learn the recent states from text stream.

There are two facts inspiring HS-PRIOR-LDA. (1) The topics in the document may contain history-state-like topics. As an example, the tweet "Libyan rebel chief gunned down in Benghazi (2011-07-28)" contains the topic that is similar to the *Military*'s history state. (2) The history-state-like topics can reuse the information stored in the history states. The word *libyan* in the aforementioned example tweet ranks much higher in the *Military* and the *Middle East* history states than the other categories' history states. After considering the relatedness between the remaining context and the history states, the learned topics of the example tweet include *Military* and *Middle East*.

The generative process of HS-PRIOR-LDA can be described as follows.

1. Draw corpus prior distribution $\boldsymbol{m} \sim Dir(\alpha \boldsymbol{u})$, where $\boldsymbol{u}$ is the uniform distribution.
2. For each topic $k \in \{1, \cdots, K\}$,
   (a) word distribution on the topic $\boldsymbol{\phi_k} \sim Dir(\boldsymbol{\beta} + \boldsymbol{\tau_k})$.
3. For each document index $d \in \{1, \cdots, D\}$,
   (a) topic distribution on the document $\theta_d \sim Dir(\boldsymbol{m})$,
   (b) for each word index $n \in \{1, \cdots, N_d\}$,
      i. word's topic assignment $z_{dn} \sim Multinomial(\theta_d)$,
      ii. word $w_{dn} \sim Multinomial(\phi_{z_{dn}})$.

In the above generative process, the line 2(a) is the key point to distinguish HS-PRIOR-LDA from LDA(Blei *et al.* 2003), where $\boldsymbol{\tau_k}$ is defined by Equation(1). K_{KB} is the number of pre-defined history-state-like topics, and S_k is the set of words appeared in the k-th topic's history state. As (Wallach 2008) mentioned that the asymmetric prior distribution can significantly improve the quality of topic modelling, $\boldsymbol{\phi_k} \sim Dir(\boldsymbol{\beta} + \boldsymbol{\tau_k})$ incorporates history state into the asymmetric prior of the word distribution on topic. The effect of $\boldsymbol{\tau_k}$ is obvious, e.g., $\tau_{Military,army}/\tau_{Military,basketball} = 203$ leads to that topic *Military* prefers to contain the word *army* other than *basketball*. The parameter λ controls how much the learned topics are similar to the history state, which can be chosen by cross-validated grid-search.

$$\tau_{kv} = \begin{cases} \lambda \dfrac{h_{kv}}{\sum_{v \in S_k} h_{kv}}, v \in S_k \; and \; k \leq K_{\boldsymbol{KB}} \\ 0, v \notin S_k \; or \; k > K_{\boldsymbol{KB}} \end{cases} \quad (1)$$

To solve HS-PRIOR-LDA, the gibbs sampling is adopted to determine the hidden variable z_{dn} and the model parameter ϕ_k. In the initialization phase of Gibbs sampling for HS-PRIOR-LDA, the hidden variable z_{dn} is initialized to topic k with probability $\hat{q}_{k|v}$ as Equation (2). For the word v that belongs to any history state, $\hat{q}_{k|v}$ is proportional to its importance in the history state τ_{kv} as Eq2(a). For the new word v in text stream, $\hat{q}_{k|v}$ is set uniformly $1/(K - K_{\boldsymbol{KB}})$ on the other topics as Eq2(b,c). The initialization makes sure that the learned topics are aligned to the pre-defined history states.

$$\hat{q}_{k|v} = \begin{cases} \dfrac{\tau_{kv}}{\sum_{k=1}^{K} \tau_{kv}}, \sum_k \tau_{kv} > 0 & (a) \\ 0, \sum_k \tau_{kv} = 0 \; and \; k \leq K_{\boldsymbol{KB}} & (b) \\ 1/(K - K_{\boldsymbol{KB}}), \sum_k \tau_{kv} = 0 \; and \; k > K_{\boldsymbol{KB}} & (c) \end{cases} \quad (2)$$

The sampling process uses the conditional probability $p(z_{dn} = k|.) \propto (n_{dk}^{(d)} + \alpha m_k)(n_{kv}^{(w)} + \tau_{kv} + \beta)/(n_{k,.}^{(w)} + \tau_{k,.} + V\beta)$, where $n_{dk}^{(d)}$ is the number of words in document d assigned to topic k, and $n_{kv}^{(w)}$ is the times of word v assigned to topic k. We also optimize $\alpha \boldsymbol{m}$ for promoting HS-PRIOR-LDA's fitting to documents according to (Wallach 2008)[4].

After Gibbs sampling on discrete time windows, HS-PRIOR-LDA learned all hidden topics of words in microblog stream. And for a specific word type w and its history-state-like topic assignment c in time window t, its document frequency is counted as $n(c, t, w^{(c)})$, which is the element of category c's t-th recent state. Event detection on recent states is discussed in the following subsection.

[4] https://github.com/mimno/Mallet/blob/master/src/cc/mallet/types/Dirichlet.java

3.3 Detecting Events from Recent States

Given the words' accurate meaning, events are detected accurately as the patterns of words' time series are better exposed. Taking the word *hood* as an example, it belongs to the category *Military* when it appeared in *Ft Hood*, a US military establishment located in Texas. As shown in Figure 1, the bursty pattern of *hood* is clearly exposed after distinguishing its *Military* time series from the raw time series.

We take the detection in three sub-phases: (1) detecting events' candidate words; (2) generating event phrases; (3) linking event tweets with event phrases.

Definition 3 (The Set of Events' Candidate Words) *The set of events' candidate words $\mathcal{B}_{c,t}$ are defined by the bursty words in recent state $\boldsymbol{r}_{c,t}$.*

Definition 4 (Event Phrase) *Event phrase $\mathcal{C}_{c,t,i}$ is the i-th combination of words which occurred in the set of events' candidate words $\mathcal{B}_{c,t}$, and represents the i-th event happened in time t under the category c.*

For a specific word w and the category c, as stream states have provided the time series $\{n(c, t_1, w^{(c)}), \cdots, n(c, t_T, w^{(c)})\}$, many bursty detection methods on the time series can be applied to check if the word w is bursty in category c.

4 Experiments

4.1 Effects of History States and Recent States

In this subsection, we demonstrate the effectiveness of *history states* initialized from knowledge base and *recent states* learned by transfer learning.

Knowledge Base. We construct the taxonomy graph $G^{(0)}$, the category-page bipartite graph $G^{(1)}$ from the latest dump of category links[5] and the page-content map $G^{(2)}$ from Wikipedia pages[6]. We set $K_{KB} = 100$, which means 100 categories are selected manually to cover the topics of Wikipedia and the target corpus as widely as possible. There are two kinds of categories are considered. The mid-high categories in the taxonomy graph $G^{(0)'}$, which are representative, are likely to be selected, such as *Aviation*, *Military*, and *Middle East*, etc. And the mid categories, which reflect the main interests of the target corpus, are also taken into consideration, such as *American Football*, *Basketball*, and *Baseball*, etc.

Text Stream Dataset. We conduct the empirical analysis on a text stream benchmark *Edinburgh twitter corpus* which is constructed by (Petrović *et al.* 2012) and widely used by previous event detection researches (Petrović *et al.* 2013) (Wurzer *et al.* 2015). Due to the developer policy of Twitter, (Petrović *et al.* 2012) only redistributes tweets' IDs[7]. We collected the tweets' contents according to the IDs with the help of Twitter API. Though we cannot get the whole dataset due to the limit of Twitter API, after necessary pre-processing, our rebuilt dataset still contains 36,627,434 tweets, which spans identically from 2011/06/30 to 2011/09/15. More details of the original dataset are described in (Petrović *et al.* 2010a).

#group*	#word*	words	NPMI
0	6-10	airlines aviation flying pilot squadron	0.113
1	11-15	flights pilots raf airways fighter	0.155
2	16-20	boeing runway force crashed flew	0.092
3	21-25	airfield landing passengers plane aerial	0.179
4	26-30	bomber radar wing bombers crash	0.137
5	31-35	airbus airports operations jet helicopter	0.189
6	36-40	squadrons base flown havilland crew	0.088
7	41-45	combat luftwaffe aerodrome carrier fokker	0.159
8	46-50	planes fly engine takeoff fleet	0.186
9	51-55	fuselage helicopters aviator naval aero	0.157
10	56-60	glider command training balloon faa	0.166
...	...	...	...
18	96-100	scheduled carriers military curtiss biplane	0.131
19	101-105	accident engines iaf albatross rcaf	0.068

Table 1: Top words' topic coherence of *Aviation*'s history state. * means the group also contains the five top words *aircraft*, *air*, *airport*, *flight*, and *airline*, but we don't show them in table to save space. NPMI is computed on ten words (a combination of words in each row with the five top words).

History States and Recent States. The history states are initialized on the pre-defined categories on the knowledge base as Algorithm 1. For HS-PRIOR-LDA, K is set to be 200, which means HS-PRIOR-LDA learned 100 history-state-like topics and 100 other topics. After cross-validated grid-search, λ is set to be 12.8. The other parameters are set as $\alpha = 0.1$, $\beta = 0.005$. We run HS-PRIOR-LDA window by window on text stream, and learn specific categories' recent states.

We compare the topic coherence of history states with the topics learned from Wikipedia by LDA in terms of NPMI(Röder *et al.* 2015). Different from traditional experiments that only compute the topic coherence of top words, we want to check whether it can hold for more words. Due to the limit of NPMI computing module[8], which computes the coherence of up to 10 words each time, we compute NPMI on the combination of top five words with each next five words as Table 1. Observe that even for the combination of 96th to 100th words with the top five words in *Aviation*'s history state, NPMI=0.131 shows that the topic coherence still holds well without drifting. More generally, Figure 4 illustrates that the history states are much more stable than the topics learned by LDA. Taking the group 10 (the combination of 56th to 60th words with the top five words) as an example, the median one of history states performs better than all those learned by LDA.

Since the history states extracted from knowledge base is stable and topic coherence,

4.2 Effects of Event Detection

Baselines Methods. We compare our proposed method against the following methods, Twevent(Li *et al.* 2012a),

[5] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-categorylinks.sql.gz

[6] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

[7] http://demeter.inf.ed.ac.uk/cross/docs/fsd_corpus.tar.gz

[8] https://github.com/AKSW/Palmetto

Aviation		*Health*		*Middle East*		*Military*		*Mobile Phones*		*Video Games*	
Normal States	Topical Words	Normal States	Topical Words	Normal States	Topical Words	Normal States	Topical Words	Normal States	Topical Words	Normal States	Topical Words
aircraft	air(3253)	health	weight(16344)	al	*#syria*(4212)	army	killed(4055)	android	iphone(13674)	game	games(8812)
air	plane	patients	loss	israel	*#bahrain*	military	news	mobile	apple	player	liked
airport	flight	medical	diet	iran	people	air	*#libya*(3503)	nokia	android	playstation	free
flight	time	disease	health	arab	israel	command	libya	ios	app	gameplay	xbox
airline	airlines	treatment	cancer	israeli	police	force	rebels	phone	ipad	nintendo	360
airlines	news	hospital	lose	egypt	*#libya*(2557)	regiment	people	samsung	samsung	games	playing
aviation	boat	patient	fat	egyptian	#egypt	forces	police	game	mobile	players	played
flying	airport	clinical	tips	ibn	news	squadron	war	app	blackberry	xbox	iphone(2820)
pilot	force	symptoms	treatment	jerusalem	#israel	infantry	libyan	iphone	tablet	mode	time
squadron	fly	cancer	body	syria	world	battle	attack	htc	apps	arcade	*ps3*

Table 2: History States extracted from Wikipedia and topical words learned from HS-Prior-LDA

Table 3: Events about *military* detected by systems between 2011-07-22 and 2011-07-28

Date	Event key words	Representative event tweet	Number of event tweet
7/22/11	Norway, Oslo, attacks, bombing	Terror Attacks Devastate Norway: A bomb ripped through government offices in Oslo and a gunman...http://dlvr.it/cLbk8	557
7/23/11	Gunman, rink	Gunman Kills Self, 5 Others at Texas Roller Rink http://dlvr.it/cLcTH	43
7/26/11	Kandahar, mayor, suicide, attack	TELEGRAPH]: Kandahar mayor killed by Afghan suicide bomber: The mayor of Kandahar, the biggest city in south _	25
7/28/11	Ft., Hood, attack	Possible Ft. Hood Attack Thwarted http://t.co/BSJ33hk	20
7/28/11	Libyan, rebel, gunned	Libyan rebel chief gunned down in Benghazi http://sns.mx/prfvy1	44

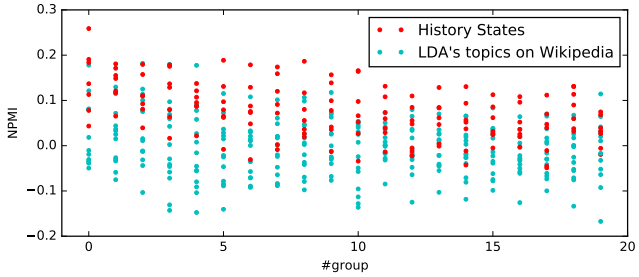

Figure 4: Effectiveness of history state in terms of NPMI

BurstyBTM(Yan *et al.* 2015b), LSH(Petrovic *et al.* 2010b), EDCoW (Weng *et al.* 2011), TimeUserLDA(Diao *et al.* 2012), and UMass System(Allan *et al.* 2000), which are mentioned in the Section 2. We implement these competing methods based on the open source community versions, e.g. EDCoW[9], or the authors' releases, e.g. BurstyBTM[10].

Evaluation Metrics. The first benchmark on *Edinburgh twitter corpus* contains 27 manually labeled events(Petrović *et al.* 2013)[11], which all exist in our rebuilt dataset on the *Edinburgh twitter*'s IDs. These labeled events focus on the events that are both mentioned in twitter and newswire,e.g. *"Oslo Attacks"* and *"US Increasing Debt Ceiling"*, but still miss many important events such as *"Hurricane Irene"*, *"Al-Qaida's No. 2 Leader Being Killed"*, and popular events such as *"Harry Potter and the Deathly Hallows (Part 2)"*. To enlarge the ground truth of realistic events, we manually evaluate the candidate events detected by LTDetector, Twevent, EDCoW, BurstyBTM, and TimeUserLDA, and use the labeled events as the second benchmark. We use precision and recall to evaluate each method on both benchmarks.

Method	Recall@ Benchmark1	Precision@ Benchmark2	Recall@ Benchmark2	DERate (Duplicate Event Rate)
UMass System	0.882	0.138	0.941	0.071
LSH	0.824	0.095	0.803	0.302
TimeUserLDA	0.353	0.536	0.071	0.054
Twevent	0.824	0.697	0.641	0.113
EDCoW	0.412	0.756	0.119	0.290
BurstyBTM	0.647	0.809	0.170	0.045
LTDetector	1.000	0.894	0.950	0.042

Table 4: Overall Performance on Event Detection

Overall Performances. Table 1 reports the number of events detected, the precision and recall, of the three methods respectively. The results of EDCoW are reproduced from [21]9 . Shown in the table, our proposed method Twevent yields the best precision of 86.1% which is significantly larger than the precisions achieved by EDCoW and Tweventu. Observe that our method Twevent detects 101 events with a recall of 75 realistic events. On the same dataset, EDCoW detects 21 events in total with 13 realistic events. Tweventu yields a slighter worse precision than EDCoW (75.3vs 76.2%) but detects the largest number of realistic events. In terms of DERate, Twevent achieves the lowest rate despite that our method detects much more events than EDCoW (101 vs 21). On the other hand, we observe that Tweventu delivers the worst DERate, more than double of Twevent (41% vs 16.0%). That is, the unigrams about the

[9] https://github.com/Falitokiniaina/EDCoW

[10] https://github.com/xiaohuiyan/BurstyBTM

[11] http://demeter.inf.ed.ac.uk/cross/docs/Newswire_Events.tar.gz

same event are clustered into two or more events. Because a tweet segment usually conveys very specific information, the tweets containing the tweet segment are all about the same topic (e.g., the event). Two tweet segments about the same event are therefore have higher chance to be clustered together.

Discussions. These methods work in four ways. 1) Based on topic model, BurstyBTM and TimeUserLDA detect bursty topics. Although they are designed for batch learning initially, we run them on the slide windows to adapt to the text stream environment. 2) Integrated with knowledge base, Twevent detects bursty segments, groups them into clusters, and then filters out the meaningless ones by Wikipedia. 3) By clustering articles, UMass System and LSH decide whether the article belongs to the existing event or a new one. As designed for First Story Detection task (one of the Topic Detection and Tracking subtasks(Allan 2012)) initially, they prefer to enlarge the recall of detected events. 4) By analyzing word frequencies, EDCoW clusters the bursty words into events.

5 Conclusions & Future Work

Knowledge base is constructed elaborately and contains rich information, which can benefit the not-well-organized text stream. As a part of our future work, we will explore the effects of transfer learning from knowledge base to text stream for more tasks, such as text classification and key words extraction, especially for short texts.

References

James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. Topic detection and tracking pilot study final report. 1998.

James Allan, Victor Lavrenko, Daniella Malin, and Russell Swan. Detections, bounds, and timelines: Umass and tdt-3. In *Proceedings of topic detection and tracking workshop*, 2000.

James Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media, 2012.

Farzindar Atefeh and Wael Khreich. A survey of techniques for event detection in twitter. *Computational Intelligence*, 31(1):132–164, 2015.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

Allan Bell. *The language of news media*. Blackwell Oxford, 1991.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *JMLR*, 2003.

Jeen Broekstra and Arjohn Kampman. Inferencing and truth maintenance in rdf schema. *PSSS*, 89, 2003.

Damian Bursztyn, François Goasdoué, Ioana Manolescu, and Alexandra Roatiş. Reasoning on web data: Algorithms and performance. In *2015 IEEE 31st International Conference on Data Engineering*, pages 1541–1544. IEEE, 2015.

Helen Caple and Monika Bednarek. Delving into the discourse: Approaches to news values in journalism studies and beyond. *Reuters Institute for the Study of Journalism, The University of Oxford, Oxford*, 2013.

Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim. Finding Bursty Topics from Microblogs. *ACL*, pages 536–544, 2012.

Stefano Faralli, Giovanni Stilo, and Paola Velardi. Large scale homophily analysis in twitter using a twixonomy. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2334–2340. AAAI Press, 2015.

Jiewen Huang, Daniel J Abadi, and Kun Ren. Scalable sparql querying of large rdf graphs. *Proceedings of the VLDB Endowment*, 4(11):1123–1134, 2011.

Jiajia Huang, Min Peng, Hua Wang, Jinli Cao, Wang Gao, and Xiuzhen Zhang. A probabilistic method for emerging topic tracking in microblog stream. *World Wide Web*, pages 1–26, 2016.

Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. 2006.

Chenliang Li, Aixin Sun, and Anwitaman Datta. Twevent: segment-based event detection from tweets. *CIKM*, pages 155–164, 2012.

Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. TEDAS: A Twitter-based Event Detection and Analysis System. *ICDE*, pages 1273–1276, 2012.

Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In *ACL*, 2013.

Ying Liu, Han Tong Loh, and Aixin Sun. Imbalanced text classification: A term weighting approach. *Expert systems with Applications*, 36(1):690–701, 2009.

Michael Mathioudakis and Nick Koudas. TwitterMonitor: trend detection over the twitter stream. *SIGMOD*, pages 1155–1158, 2010.

Thien Huu Nguyen and Ralph Grishman. Event detection and domain adaptation with convolutional neural networks. In *ACL*, 2015.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Saša Petrović, Miles Osborne, and Victor Lavrenko. The edinburgh twitter corpus. In *NAACL-HLT*, 2010.

Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Streaming First Story Detection with application to Twitter. *HLT-NAACL*, pages 181–189, 2010.

Saša Petrović, Miles Osborne, and Victor Lavrenko. Using paraphrases for improving first story detection in news and twitter. In *NAACL-HLT*, 2012.

Saša Petrović, Miles Osborne, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Can twitter replace newswire for breaking news? In *ICWSM*, 2013.

Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *WSDM*, 2015.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. Sentiment in twitter events. *Journal of the American Society for Information Science and Technology*, 62(2):406–418, 2011.

Hanna Megan Wallach. *Structured topic models for language*. PhD thesis, University of Cambridge, 2008.

Jianshu Weng, Yuxia Yao, Erwin Leonardi, and Francis Lee. Event Detection in Twitter. *ICWSM*, pages 1–22, July 2011.

Dominik Wurzer, Victor Lavrenko, and Miles Osborne. Twitter-scale New Event Detection via K-term Hashing. In *EMNLP*, 2015.

Rui Yan, Yiping Song, Cheng-Te Li, Ming Zhang, and Xiaohua Hu. Opportunities or Risks to Reduce Labor in Crowdsourcing Translation? Characterizing Cost versus Quality via a PageRank-HITS Hybrid Model. *IJCAI*, pages 1025–1032, 2015.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. A Probabilistic Model for Bursty Topic Discovery in Microblogs. *AAAI*, pages 353–359, 2015.

Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420, 1997.

Jie Yin, Sarvnaz Karimi, Bella Robinson, and Mark A Cameron. ESA: emergency situation awareness via microbloggers. *CIKM*, pages 2701–2703, 2012.