Category-Level Transfer Learning from Knowledge Base to Microblog Stream for Accurate Event Detection

Weijing Huang, Tengjiao Wang, Wei Chen(✉), and Yazhou Wang

Key Laboratory of High Confidence Software Technologies (Ministry of Education), EECS, Peking University, Beijing, China
huangwaleking@gmail.com, {tjwang,pekingchenwei}@pku.edu.cn, pkuwangyz@gmail.com

Abstract. Many Web applications need the accurate event detection technique on microblog stream. But the accuracy of existing methods is still challenged by microblog's short length and high noise. We develop a novel category-level transfer learning method TransDetector to deal with the task. TransDetector bases on two facts, that microblog is short but can be enriched by knowledge base semantically with transfer learning; and events can be detected more accurately on microblogs with richer semantics. The following contributions are made in TransDetector. (1) We propose a structure-guided category-level topics extraction method, which exploits the knowledge base's hierarchical structure to extract categories' highly correlated topics. (2) We develop a probabilistic model CTrans-LDA for category-level transfer learning, which utilizes the word co-occurrences and transfers the knowledge base's category-level topics into microblogs. (3) Events are detected accurately on category-level word time series, due to richer semantics and less noise. (4) Experiment verifies the quality of category-level topics extracted from knowledge base, and the further study on the benchmark *Edinburgh twitter corpus* validates the effectiveness of our proposed transfer learning method for event detection. TransDetector achieves high accuracy, promoting the precision by 9% without sacrificing the recall rate.

Keywords: Event detection · Microblog stream · Transfer learning · Knowledge base

1 Introduction

Many Web applications need the accurate event detection technique on microblog stream, such as public opinion analysis [1], public security [2], and disaster response [3], etc. Although event detection has been a research topic

This research is supported by the Natural Science Foundation of China (Grant No. 61572043), and the National Key Research and Development Program (Grant No. 2016YFB1000704).

S. Candan et al. (Eds.): DASFAA 2017, Part I, LNCS 10177, pp. 50–67, 2017.
DOI: 10.1007/978-3-319-55753-3_4

for a long while [4], event detection in microblog stream is still challenging [5]. According to [6], the characteristics of microblog, which are fast changing, high noise, and short length, raise the challenge.

Knowledge base can be a good supplementary for event detection on microblog stream to address these challenges. 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*" at a higher conceptual level. By regarding these wiki information and the word *attack* (also highly related to *Military*), the model easily understands the example tweet is about something related to *Military*. In other words, knowledge base enriches the linkages between words and concepts, and provides more comprehensive context for microblogs. Since the transfer learning [7] aims at utilizing the extra information stored in the source dataset to benefit the target dataset, it provides a feasible way to enhance the event detection in microblog stream. Taking Fig. 1 below as an example, the fluctuation of time series of raw word *hood* is not so obvious to reflect the event happened to the military base *Ft. Hood*. After transferring the knowledge about *"Military"* into the microblog stream, the time series of word *"hood"* related to *"Military"* is extracted, which is more vivid to detect the event happened on July 28th, 2011.

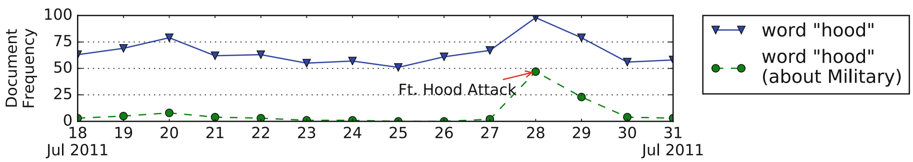


Fig. 1. The comparison of the time series between the raw word *hood* and the *Military* related word *hood*, computed on the *Edinburgh twitter corpus* [8]. The rise of document frequency on July 28th, 2011 is corresponding to the event mentioned in https://en.wikipedia.org/wiki/Fort_Hood#2011_attack_plot.

The benefit of enriching the semantics for micorblogs is attractive, but it's non-trivial to transfer the knowledge stored in the knowledge base into microblog stream directly. The existing RDF model [9] 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 [10], which is not suitable for the scene of quickly and accurately detecting events. In existing methods, what meets the demand most is Twevent [11], but it's limited in only treating Wikipedia as a looking-up table and may drop some events incorrectly.

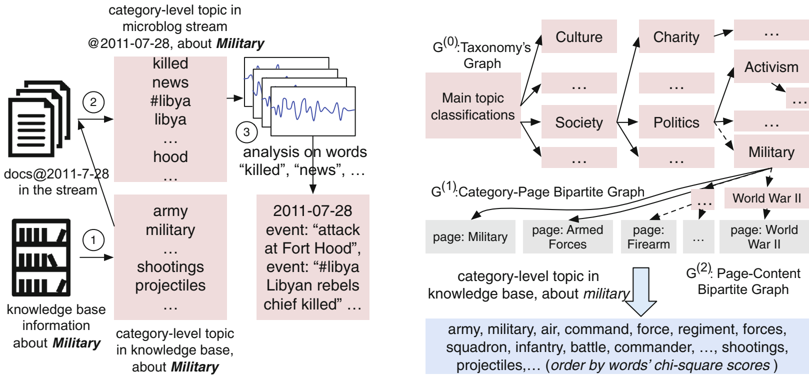
In our paper, to balance the performance and the cost of leveraging knowledge base for event detection, we develop a novel category-level transfer learning method, namely TransDetector. It consists of three phases: extracting category-level topics in knowledge base, conducting transfer learning to get category-level topics in microblog stream, and detecting events from category-level word time series, as illustrated in Fig. 2(a). We explain the main idea of the three parts in TransDetector, and leave more technique details in Sect. 3. (1) In the **extracting** phase, we propose a structural-guided category-level topics extraction method on the knowledge base. It bases on the following facts. The knowledge base has the three fold hierarchical structure, consisting of the taxonomy graph (*class* $\rightarrow$ *subclass*), the category-page bipartite graph (*class* $\rightarrow$ *instance*), and the page-content map (*instance* $\rightarrow$ *content*), as illustrated in Fig. 2(b). 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, and restore them in *category-level topics in knowledge base*. (2) In the **conducting transfer learning** phase, we propose a novel probabilistic model CTrans-LDA for transferring the knowledge. CTrans-LDA works in the bayesian transfer learning way like [12], by utilizing the extracted *category-level topics in knowledge base* as the informative priors to bridge two data domains. CTrans-LDA labels whether a word in a microblog message should link to a category in knowledge base, or just label it as no-category-related word. Applying CTrans-LDA on more microblogs, it gets the *category-level topics in microblog stream* and *category-level word time series*. (3) In the **detecting** phase, since the words in microblogs have been enriched semantically and the meaningless words are labeled as no-category-related, the event detection on *category-level word time series* is much more accurate than other methods which don't conduct the transfer learning. And the experiment on the *Edinburgh twitter corpus* demonstrates the effectiveness of our proposed TransDetector.

To sum up, the contribution of this paper is mainly in four aspects. (1) We propose a structure-guided category-level topics extraction method, exploiting the knowledge base's hierarchical structure to extract categories' highly correlated topics. (2) We develop a probabilistic model CTrans-LDA for category-level transfer learning, which utilizes the word co-occurrences and transfers the knowledge base's category-level topics into microblogs. (3) Events are detected accurately on category-level word time series, due to richer semantics and less noise. (4) Experiment verifies the quality of category-level topics extracted from knowledge base, and the further study on the benchmark *Edinburgh twitter corpus* validates the effectiveness of our proposed transfer learning method for event detection.

2 Related Works

Event Detection. Based on how much data are used, the methods are mainly in two groups, *without extra information* and *leveraging extra information*.

Most existing methods are implemented *without extra information*. They are mainly carried out by clustering articles [13–15], analyzing word frequencies



(a) TRANSDETECTOR's Process Flow (b) Extracting Category-Level Topics in Knowledge Base

Fig. 2. (a) TRANSDETECTOR's process flow, taking *Military* related events in microblogs as an example. TRANSDETECTOR includes three phases: extracting category-level topics in knowledge base, conducting transfer learning to get category-level topics in microblog stream, and detecting events from category-level word time series; (b) Illustration on how to extract category-level topics in knowledge base via its three fold hierarchical structure, taking *Military* as an example.

[16,17], or finding bursty topics [18,19] etc. (1) By clustering articles, UMass [13], LSH [14], k-Term-FSD [15] and other similar methods model the occurred events as clusters of articles. The decision is based on whether the dissimilarity between the incoming article and existed event clusters is over the user-specified threshold. (2) By analyzing word frequencies, EDCoW [17] exploits wavelet transformation, which converts signal from time domain to time-scale domain, to detect the change point of word signal. However they treat the word as the most basic unit in analysis, without regarding polysemy words. (3) By finding bursty topics via topic modelling, such as TimeUserLDA [18], EUTB [20], and BurstyBTM [19]. TimeUserLDA and EUTB distinguish user's long term interests and short term bursty events, and BurstyBTM utilizes the burstiness of biterms as prior knowledge. This kind method needs to set the appropriate number of topics, and detects the "large" events but ignore the "small" ones. To summarize, due to microblog's short length and high noise, it's not easy to set the directly applicable parameter for existing methods to achieve high precision and high recall simultaneously.

The second group methods detect events by *leveraging extra information.* [21] incorporates user's and the followers' profiles to help to detect the events of public concern. However it's not easy to get this kind contextual information. [22] compares two time series generated by event-related tweets and the corresponding Wikipedia article's page views, and further filter out spurious events. Twevent [11] divides the tweet into segments according to the Microsoft Web N-Gram service and Wikipedia, then detect the bursty segments and cluster these segments into candidate events for necessary post processings. But as shown in [19], Twevent is still hampered by the performance of simply clustering the bursty segments. Beyond the scope of microblog stream, [23] utilizes the concurrent wikipedia edit spikes for event detection. In a nutshell, existing works are different from ours, as we transfer the category information of knowledge base into microblog's words and get the finer processing objects in event detection, rather than treating the knowledge base as a lookup table or a comparison base.

Knowledge Base. Many general knowledge bases and customized knowledge bases are constructed and utilized for different text mining tasks. For example, Probase [24] constructs a large general probabilistic *IsA* taxomomy from webpages, and is used for semantic web search, and text classification [25]. Such kind efforts are also made by DBPedia [26], Yago [27], and Freebase [28]. These works manages knowledge as tuples on graph. However the query on large graph is very expensive [10], which is not very suitable for the scene of quickly and accurately detecting events. The customized knowledge bases such as EVIN [29], Event Registry [30], and Story-base [31], are designed for managing events. These knowledge bases are mainly built on news articles. EVIN maps the existing event-related news articles into semantic classes. Event Registry collects the articles by the API of News Feed Service [32], then detects events, and provides the structural information of the events, such as the related Wikipedia article, timestamp, and location etc. Storybase introduces Wikipedia current events[1] as the resources for constructing event-and-storyline knowledge base on news articles, which are provided by GDELT project [33]. As the news articles may lag behind the microblogs when the emergency of events, it's not enough to detect events from microblog stream by directly applying these customized knowledge bases.

Transfer Learning. Although the training examples in the target domain is very sparse, transfer learning [7] can utilize the domain-independent knowledge as the bridge to fill the information gap between source domain and target domain [34] to get more "examples" for training. By bayesian transfer learning, [12] uses the distribution in source domain as the prior knowledge for text classification. And inspired by this idea, we propose the *category-level topic* as the bridge for transferring knowledge.

3 TransDetector

In this section, we present a novel category-level transfer learning method for event detection, namely TransDetector. Illustrated in Fig. 2(a), TransDetector consists of three parts: (1) extracting category-level topics in knowledge base, (2) conducting transfer learning to get category-level topics in microblog stream, and (3) detecting events from category-level word time series. The following definitions are used by our method TransDetector. Definition 1 is used in

[1] https://en.wikipedia.org/wiki/Portal:Current_events.

the **extracting** phase, Definitions 2 and 3 are used in the **conducting transfer learning** phase, and Definitions 4 to 6 are used in the **detecting** phase. Because Definitions 2 to 6 are very easy to understand, what needs to be specified are Definition 1. *Category-level topics in knowledge base* is a set of tuples extracted from knowledge base, which weighs the importance of given words to the specific category by the chi-square score, defined in Definition 1 formally. Taking the *Military* category in Fig. 2(b) as an example, the category-level topic of *military* contain the words *army*, *military*, and *shootings* etc., which are highly related to the category *Military*.

Definition 1 (Category-Level Topics in Knowledge Base). Given category c, the category level topic in knowledge base 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 c's category-level topic in knowledge base as $\boldsymbol{h}_c = \{< w_i^{(c)}, chi(c, w_i^{(c)}) >\}_{i=1}^{N_c}$.

Definition 2 (Category-Level Topics in Microblog Stream). Given category c, the category level topic in microblog stream at time t is defined by a set of tuples, and denoted as $\boldsymbol{r}_{c,t} = \{< w_i^{(c)}, n(c, t, w_i^{(c)}) >\}_{i=1}^{N_c}$, 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.

Definition 3 (Category-Level Word Time Series). Given category c and word w, the category-level word time series is a list of document frequencies extracted from $\{\boldsymbol{r}_{c,t}\}_{t=1}^{T}$, and denoted as $\{n(c, t, w_i^{(c)})\}_{t=1}^{T}$.

Definition 4 (The Set of Events' Candidate Words). The set of events' candidate words $\mathcal{B}_{c,t}$ are groups of the bursty words in $\boldsymbol{r}_{c,t}$.

Definition 5 (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.

Definition 6 (Event Related Microblogs). Event related microblogs $\mathcal{D}_{c,t,i}$ are articles relating to the i-th event in time t of category c, and correspond to the event phase $\mathcal{C}_{c,t,i}$.

Section 3.1 explains the details of how to extract *category-level topics in knowledge base* by using the three fold hierarchical structure. Section 3.2 shows how to conduct transfer learning to get *category-level word time series* in microblog stream. And Sect. 3.3 interprets the high accuracy event detection based on the processed *category-level word time series*.

3.1 Extracting Category-Level Topics in Knowledge Base

In this part, we discuss in detail how to extract the *category-level topics in knowledge base* on the given categories. 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 [9], which is adopted to build DBPedia [26] and YAGO [27] from Wikipedia. But the the query on the large graph is usually very expensive [10]. To make a trade-off between cost and performance, we use the lightweight data structure *topics* to represent the knowledge at category(class) level. And the knowledge base's threefold structure $G^{(0)}$, $G^{(1)}$, $G^{(2)}$ benefits the extraction of *category-level topics*.

Taxonomy Graph $G^{(0)}$. The directed edges in $G^{(0)}$ represent the *class→subclass* relations in the knowledge base. Taking Wikipedia for example (Fig. 2(b)), the node *Main topic classifications*[2] 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 [35], we remove the cycles according to nodes' PageRank-HITS [36] 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.

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 information stored 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 [37,38] 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, we can get the *category-level topic*, which are composed by the category related words and their corresponding importance.

Considering the full Wikipedia's contents, the word's score in the *category-level topic* evaluates the importance of word to the concerned category accurately. For example, in *Military*'s *category-level topic*, 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.

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

Algorithm 1. Extraction of Category-Level Topics in 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: c's category-level topic in knowledge base $\boldsymbol{h}_c$

$Pages(c) \leftarrow \varnothing$, $\boldsymbol{h}_c \leftarrow \varnothing$
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 $\boldsymbol{h}_c$

3.2 Conducting Transfer Learning

In this subsection, we describe how the proposed probabilistic model CTrans-LDA transfers the category-level topics into microblogs.

There are two facts inspiring CTrans-LDA. (1) The topics in the document may contain category-level 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 *category-level topic in knowledge base*. (2) The *category-level topics in microblog stream* can reuse the information stored in the corresponding *category-level topics in knowledge base*. The word *libyan* in the aforementioned example tweet ranks much higher in the *Military* and the *Middle East* category-level topics than the other categories' topics. After considering the relatedness between the remaining context and the *category-level topics in knowledge base*, the learned topics of the example tweet include *Military* and *Middle East*. The generative process of CTrans-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 is the key point to distinguish CTrans-LDA from LDA [39], where $\boldsymbol{\tau}_k$ is defined by Eq. (1). K_{KB} is the number of pre-defined categories used for transferring, and S_k is the set of words appeared in the k-th category-level topic. As [40] 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 *category-level topics in knowledge base* 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 knowledge is used for transferring, which can be chosen by cross-validated grid-search.

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

To solve CTrans-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 CTrans-LDA, the hidden variable z_{dn} is initialized to topic k with probability $\hat{q}_{k|v}$ as Eq. (2). For the word v that belongs to any category-level topic, $\hat{q}_{k|v}$ is proportional to its importance in the category-level topic τ_{kv} as Eq. 2(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 Eq. 2(b, c). The initialization makes sure that the learned topics are aligned to the pre-defined category-level topic.

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

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 set $\alpha = 0.1$ and $\beta = 0.005$ according to the suggestion by [41].

After Gibbs sampling on discrete time windows, CTrans-LDA learned all hidden topics of words in microblog stream. And for a specific word type w and its category assignment c in time window t, its document frequency is counted as $n(c, t, w^{(c)})$, which is the element of category-level word time series. Event detection on category-level word time series is discussed in the following subsection.

3.3 Detecting Events from Category-Level Word Time Series

After transfer learning, it's much easier to detect the event from category-level word time series, which is shown in Fig. 1. We take the detection in three sub-phases: (1) detecting events' candidate words; (2) generating event phrases; and (3) retrieving event related microblogs. Note that these sub-phases are also available for the common text stream, but they perform better when words are enriched by category concepts of knowledge base.

Detecting events' candidate words. For a word w and a category c, on its time series $\{n(c,t,w^{(c)})\}_{t=1}^{T}$, many bursty detection methods can be applied to check if the word w is bursty in category c by given time t. In our paper, we assume the document frequency of $w^{(c)}$ follows a poisson distribution, which is also used in [18]. And the document frequency of nonbursting $w^{(c)}$ has the poisson distribution with the parameter $\lambda = \mu_0$; while the bursting one with the parameter $4\mu_0$ means the document frequency $n(c,t,w^{(c)})$ is much higher when bursting. We empirically set μ_0 as the moving average $\mu_0^{(t)} = \frac{1}{T}\sum_{\tau=t-T}^{t-1} n(c,\tau,w^{(c)})$, and set T=10. Hence, at time t, bursty or not is determined by comparing the probabilities of above poisson distributions. After bursty detection on the time series, we add each time t's bursty word $w^{(c)}$ into the set of event's candidate words $\mathcal{B}_{c,t}$.

Generating event phrases. In the set of event's candidate words $\mathcal{B}_{c,t}$, some words appear together and should be grouped into event phrases. For example, there are six bursty words in $\mathcal{B}_{military,2011-07-28}$ belonging to two event phrases *"ft, hood, attack"* and *"libyan, rebel, gunned"* in the time window 2011-07-28 to represent the events happened to the US military establishment and the Libyan rebel respectively.

To group the bursty words together, we construct the directed weighted graph $\mathcal{G}_{c,t} = (\mathcal{B}_{c,t}, \mathcal{E}_{c,t}, \mathcal{W}_{c,t})$, where $\mathcal{E}_{c,t} = \{(a,b)|a \in \mathcal{B}_{c,t}, b \in \mathcal{B}_{c,t}, PMI(a,b) > 0\}$, and $\mathcal{W}_{c,t}$ gives the PMI scores on the edges. The graph $\mathcal{G}_{c,t}$ means the words in $\mathcal{B}_{c,t}$ are connected if and only if their PMI score in the given time window's microblogs is over 0. Given the graph $\mathcal{G}_{c,t}$, the spectral clustering [42] is utilized for exploring the best partition of words. To get the optimum cluster number, we use the graph density as the criteria. The graph density is the ratio of the number of edges to that of complete graph (the graph with all possible edges). We search the best cluster number from 1 to $|\mathcal{B}_{c,t}|$, and stop when all the resulting subgraphs satisfy the criteria that the density is over the threshold, which is set to 0.6 empirically. In this way, the generated event phrase combines the co-occurred bursty words and excludes the unrelated.

Retrieving event related microblogs. To better understand the event, we retrieve the event microblogs $\mathcal{D}_{c,t,i}$ by using the event phrase $\mathcal{C}_{c,t,i}$. Generally, according to the number of bursting words in the event phrase $\mathcal{C}_{c,t,i}$, there are two situations to be addressed. (1) $|\mathcal{C}_{c,t,i}| = 1$, we directly add the microblog in time window t, which contains the bursting category-word $w^{(c)}$, into the set $\mathcal{D}_{c,t,i}$. (2) $|\mathcal{C}_{c,t,i}| \geq 2$, it's not necessary that all the bursting words are included in the event related microblog. For example, the tweet "*Soldier wanted to attack Fort Hood troops*" contains the bursting words *attack* and *hood*, not all the event phrase "*ft, hood, attack*". To tackle this problem, we consider the microblog, that contains any pair of category-words in the event phrase, as the event related microblog. Finally, TransDetector gets the detected events $\{(\mathcal{C}_{c,t,i}, \mathcal{D}_{c,t,i})\}_{i=1}^{|\mathcal{C}_{c,t}|}$, containing the event phrase and the corresponding event related microblogs for the given category and the time window.

4 Experiments

In this subsection, we demonstrate the effectiveness of TransDetector, by evaluating the *category-level topics in knowledge base*, *category-level topics in microblog stream* learned by transfer learning, and the effect of event detection.

Knowledge Base. We construct the taxonomy graph $G^{(0)}$, the category-page bipartite graph $G^{(1)}$ from the latest dump of category links[3] and the page-content map $G^{(2)}$ from Wikipedia pages[4]. We set $K_{KB} = 50$, which means 50 categories are selected manually to cover the topics of Wikipedia and the target corpus as widely as possible. There are two kinds of categories 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.

Microblog Stream Dataset. We conduct the empirical analysis on a text stream benchmark *Edinburgh twitter corpus* which is constructed by [8] and widely used by previous event detection researches [15,43]. Due to the developer policy of Twitter, [8] only redistributes tweets' IDs[5]. 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 also spans from 2011/06/30 to 2011/09/15. More details of the original dataset are described in [44].

4.1 Evaluation on Categroy-Level Topics in Knowledge Base

The category-level topics are initialized on the pre-defined categories from the knowledge base as Algorithm 1.

Evaluation Metrics. We compare the topic coherence of category-level topics with the topics learned from Wikipedia by LightLDA [45] in terms of NPMI [46]. 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[6], 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.

Results. Observe that even for the combination of 96th to 100th words with the top five words in *Aviation*'s category-level topics in knowledge base, NPMI = 0.131 shows that the topic coherence still holds well without drifting. More generally, Fig. 3 illustrates that the category-level topics are much more

[3] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-categorylinks.sql.gz.
[4] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2.
[5] http://demeter.inf.ed.ac.uk/cross/docs/fsd_corpus.tar.gz.
[6] https://github.com/AKSW/Palmetto.

Table 1. The comparison on the topic coherence (NPMI) between our method and LightLDA [45], taking *Aviation* as an example. (GID is short for Group Id. * means each group contains the five top words. NPMI is computed on a group of ten words. ∼ stands for the top five words.)

Category-Level Topics extracted from Wikipedia by TRANSDETECTOR				Topics Learned from Wikipedia by LightLDA[47]			
GID	#words*	words	NPMI	GID	#words*	words	NPMI
-	1-5	aircraft air airport flight airline	-	-	1-5	engine aircraft car air power	-
0	1-5, 6-10	∼, airlines aviation flying pilot squadron	0.113	0	1-5, 6-10	∼, design flight model production speed	0.112
1	1-5, 11-15	∼, flights pilots raf airways fighter	0.155	1	1-5, 11-15	∼, system vehicle cars engines mm	0.062
2	1-5, 16-20	∼, boeing runway force crashed flew	0.092	2	1-5, 16-20	∼, fuel vehicles designed models type	0.072
3	1-5, 21-25	∼, airfield landing passengers plane aerial	0.179	3	1-5, 21-25	∼, version front produced rear electric	0.035
4	1-5, 26-30	∼, bomber radar wing bombers crash	0.137	4	1-5, 26-30	∼, space control motor standard development	0.085
5	1-5, 31-35	∼, airbus airports operations jet helicopter	0.189	5	1-5, 31-35	∼, film range light using available	-0.002
6	1-5, 36-40	∼, squadrons base flown havilland crew	0.088	6	1-5, 36-40	∼, wing powered wheel weight launch	0.087
7	1-5, 41-45	∼, combat luftwaffe aerodrome carrier fokker	0.159	7	1-5, 41-45	∼, developed low test ford cylinder	0.007
8	1-5, 46-50	∼, planes fly engine takeoff fleet	0.186	8	1-5, 46-50	∼, equipment side pilot hp aviation	0.091
9	1-5, 51-55	∼, fuselage helicopters aviator naval aero	0.157	9	1-5, 51-55	∼, systems us sold body drive	-0.051
10	1-5, 56-60	∼, glider command training balloon faa	0.166	10	1-5, 56-60	∼, gear introduced class safety seat	0.069
...	...	...	...	...	...	...	...
18	1-5, 96-100	∼, scheduled carriers military curtiss biplane	0.131	18	1-5, 96-100	∼, transmission special replaced limited different	0.059
19	1-5, 101-105	∼, accident engines iaf albatross rcaf	0.068	19	1-5, 101-105	∼, features machine nuclear even unit	0.011

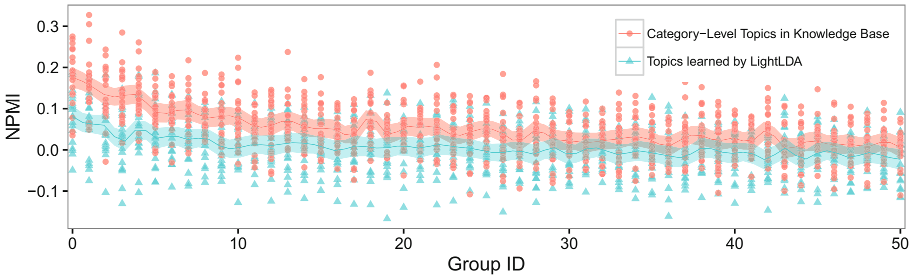


Fig. 3. More topics are compared at the NPMI metrics between our method and LightLDA

stable than the topics learned by LightLDA. Taking the group 10 (the combination of 56th to 60th words with the top five words) as an example, the median one of category-level topic performs better than those learned by LightLDA. And more topics are compared shown in Fig. 3, proving that our method performs better at extracting key features for categories.

For CTrans-LDA, K is set to be 200, which means CTrans-LDA learned 50 category-level topics and 150 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 CTrans-LDA window by window on text stream, and learn specific categories' topics. The result in Table 2 demonstrates the effectiveness of transferring category-level topics into the microblog stream, and finding more new words in the stream which is not stored in the knowledge base.

4.2 Effects of Event Detection

Baseline Methods. We compare our proposed method against the following methods, Twevent [11], BurstyBTM [19], LSH [14], EDCoW [17], and

Table 2. Category-Level Topics extracted from knowledge base and the corresponding topics on microblog stream learned from CTrans-LDA, taking the categories *Aviation*, *Health*, *Middle East*, *Military*, and *Mobile Phones* as examples. The words in ***bold italic*** font are newly learned on the microblog stream by the transfer learning, which semantic meanings are verified consistent with the categories; while the words in normal font play the role as the bridge in transfer learning, and appear in the both category-level topics in two domains.

Aviation		*Health*		*Middle East*		*Military*		*Mobile Phones*	
Knowledge Base	Microblog Stream	Knowledge Base	Microblog Stream	Knowledge Base	Microblog Stream	Knowledge Base	Microblog Stream	Knowledge Base	Microblog Stream
aircraft	air	health	weight	al	***#syria***	army	killed	android	iphone
air	plane	patients	loss	israel	***#bahrain***	military	news	mobile	apple
airport	flight	medical	diet	iran	people	air	***#libya***	nokia	android
flight	time	disease	health	arab	israel	command	libya	ios	app
airline	airlines	treatment	cancer	israeli	police	force	rebels	phone	ipad
airlines	news	hospital	lose	egypt	***#libya***	regiment	people	samsung	samsung
aviation	boat	patient	fat	egyptian	#egypt	forces	police	game	mobile
flying	airport	clinical	tips	ibn	news	squadron	war	app	blackberry
pilot	force	symptoms	treatment	jerusalem	***#israel***	infantry	libyan	iphone	tablet
squadron	fly	cancer	body	syria	world	battle	attack	htc	apps

TimeUserLDA [18], which are mentioned in the Sect. 2. We implement these competing methods based on the open source community versions, e.g. EDCoW[7], or the authors' releases, e.g. BurstyBTM[8]. The above methods are set according to the descriptions in their papers. More precisely, (1) for LSH, 13 bits per hash table, 20 hash tables are set, and top 500 clusters with high entropy are selected as the event candidates; (2) for Twevent, the number of candidate bursty segments is set to be the square root of the window size, the newsworthiness threshold is set to be 4, and 375 candidate events are detected; (3) for EDCoW, the parameter γ is set to be 40, and 349 bursty "phrases" are found for evaluation; (4) for TimeUserLDA, the topic number is set to be 500, and the most 100 bursty topics are selected as candidate events; (5) for BurstyBTM, the topic number is set to be 200, which is also the number of bursty topics in the model. The information about the number of events to be evaluated is listed in Table 3, where TransDetector detects 457 events after filtering out too niche events containing less than 20 tweets.

Benchmarks and Evaluation Metrics. The evaluation is conducted on two benchmarks. The first benchmark on *Edinburgh twitter corpus* contains 27 manually labeled events [43][9], 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 include these important events and enlarge the ground truth of realistic events pool, we build the second benchmark carefully.

[7] https://github.com/Falitokiniaina/EDCoW.
[8] https://github.com/xiaohuiyan/BurstyBTM.
[9] http://demeter.inf.ed.ac.uk/cross/docs/Newswire_Events.tar.gz.

We manually evaluate the candidate events detected by LSH, TransDetector, Twevent, EDCoW, BurstyBTM, and TimeUserLDA, with the help of the *Wikipedia Current Event Portal*[10] and a local search engine built on Lucene. The labeling process generates the Benchmark2, and contains 395 events. We use precision and recall to evaluate each method on both benchmarks, and utilize the DERate (Duplicate Event Rate) metric [11] to measure the readability of detected events. The smaller the metric DERate, the less duplicate events to be filtered out in the application.

Results. In general, our method is better than the existing methods in terms of the precision and the recall, only sacrificing in the DERate slightly, as shown in Table 3. This is because an event could be grouped into multiple categories (e.g. the event *"S&P downgrade US credit rating"*, related to the politics category and the financial category simultaneously), but is not a problem as the method TrasnDetector has already achieved the high precision and recall.

Comparing to TrasnDetector, the existing methods are suffered from having to choose between precision and recall, but not both. Taking Twevent as an example, which also utilizes the knowledge base for promoting the event detection, it has three parameters to trade off, in which the newsworthiness is the most impact one. In our experiment, when setting newsworthiness to be 4, Twevent achieves its best performance in terms of F value; lower or higher of value sacrifices the recall or precision. The other methods also meet the same problem of tuning parameters, such as the topic number for TimeUserLDA and BurstyBTM, and the distance threshold for LSH.

Table 3. Overall Performance on Event Detection

Method	Number of events to be evaluated	Recall@ Benchmark1	Precision@ Benchmark2	Recall@ Benchmark2	F@ Benchmark2	DERate[a] (Duplicate Event Rate)@ Benchmark2
LSH	500	0.704	0.788	0.651	0.713	0.348
TimeUserLDA	100	0.370	0.790	0.177	0.289	0.114
Twevent	375	0.741	0.808	0.658	0.725	0.142
EDCoW	349	0.556	0.748	0.511	0.607	0.226
BurstyBTM	200	0.667	0.825	0.384	0.497	**0.079**
TransDetector	457	**0.889**	**0.912**	**0.876**	**0.894**	0.170

[a] DERate = (the number of duplicate events)/(the total number of detected realistic events)[11]

In order to understand why the existing methods cannot perform well, we dive into the results on the Benchmark1, and show the relation between the recall and the event size (number of microblogs related to the events) in Fig. 4.

[10] https://en.wikipedia.org/wiki/Portal:Current_events.

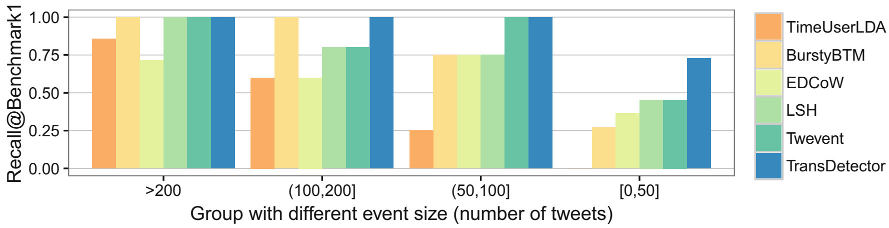


Fig. 4. The relation between the recall and the event size

Table 4. 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	Methods[a]					
				L	TU	TW	E	B	TD
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_	47	✓	-	✓	✓	-	✓
7/28/11	Ft., Hood, attack	Possible Ft. Hood Attack Thwarted http://t.co/BSJ33hk	52	-	-	-	-	-	✓
7/28/11	Libyan, rebel, gunned	Libyan rebel chief gunned down in Benghazi http://sns.mx/prfvy1	44	-	-	-	-	-	✓

[a] L = LSH [14], TU = TimeUserLDA [18], TW = Twevent [11], E = EDCoW [17], B = BurstyBTM [19], TD = TransDetector.

The 27 events are divided into 4 groups according to its size. The first group contains 5 very popular events, having more than 200 microblogs, such as *"S&P downgrade US credit rating"* with 656 microblogs, *"Atlantis shuttle lands"* with 595 microblogs, etc. The remaining groups' sizes are listed in the x-axis of Fig. 4.

The fourth group contains 11 not-so-popular events, each containing less than 50 microblogs, such as *"War criminal Goran hadzic arrested"* with only 27 microblogs reported. In our experimental settings, the methods all perform well on the very popular events, but perform worse on the not-so-popular events. For the not-so-popular events, the underlying bursty pattern is not so obvious that leads many methods to fail to catch the pattern. We further find out that the number of microblogs in an event follows a power-law distribution, which similar phenomenon is also reported in [11]. The widespread existence of such not-so-popular events challenge almost all the existing event detection methods. More result in Table 4 shows the examples of military related events detected by TransDetector on the real dataset.

5 Conclusions and Future Work

Knowledge base is constructed elaborately and contains rich information, which can benefit the not-well-organized text stream. In our proposed TransDetector method, by using category-level topic in knowledge base as the prior knowledge, we transfer abundant knowledge from knowledge base into microblog stream, which enriches the semantics of microblogs and further enhances the accuracy of microblogs event detection. 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.

Acknowledgments. Thanks to Dr. Jian Tang and the anonymous reviewers for giving valuable suggestions on this work.

References

1. Thelwall, M., Buckley, K., Paltoglou, G.: Sentiment in twitter events. J. Am. Soc. Inf. Sci. Technol. **62**(2), 406–418 (2011)
2. Li, R., Lei, K.H., Khadiwala, R., Chang, K.C-C.: TEDAS: a Twitter-based event detection and analysis system. In: ICDE (2012)
3. Yin, J., Karimi, S., Robinson, B., Cameron, M.A.: ESA: emergency situation awareness via microbloggers. In: CIKM (2012)
4. Allan, J., Carbonell, J.G., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study final report (1998)
5. Atefeh, F., Khreich, W.: A survey of techniques for event detection in twitter. Comput. Intell. **31**(1), 132–164 (2015)
6. Huang, J., Peng, M., Wang, H., Cao, J., Gao, W., Zhang, X.: A probabilistic method for emerging topic tracking in microblog stream. In: World Wide Web (2016)
7. Pan, S.J., Yang, Q.: A survey on transfer learning. TKDE **22**(10), 1345–1359 (2010)
8. Petrović, S., Osborne, M., Lavrenko, V.: Using paraphrases for improving first story detection in news and twitter. In: NAACL-HLT (2012)
9. Klyne, G., Carroll, J.J.: Resource description framework (RDF): concepts and abstract syntax (2006)

10. Huang, J., Abadi, D.J., Ren, K.: Scalable SPARQL querying of large RDF graphs. In: VLDB (2011)
11. Li, C., Sun, A., Datta, A.: Twevent: segment-based event detection from tweets. In: CIKM (2012)
12. Dai, W., Xue, G.-R., Yang, Q., Yu, Y.: Transferring naive bayes classifiers for text classification. In: AAAI (2007)
13. Allan, J., Lavrenko, V., Malin, D., Swan, R.: Detections, bounds, timelines: UMass and TDT-3. In: Proceedings of Topic Detection and Tracking Workshopp (2000)
14. Petrovic, S., Osborne, M., Lavrenko, V.: Streaming first story detection with application to twitter. In: HLT-NAACL (2010)
15. Wurzer, D., Lavrenko, V., Osborne, M.: Twitter-scale new event detection via K-term hashing. In: EMNLP (2015)
16. Mathioudakis, M., Koudas, N.: TwitterMonitor: trend detection over the twitter stream. In: SIGMOD (2010)
17. Weng, J., Yao, Y., Leonardi, E., Lee, F.: Event detection in twitter. In: ICWSM (2011)
18. Diao, Q., Jiang, J., Zhu, F., Lim, E.-P.: Finding bursty topics from microblogs. In: ACL (2012)
19. Yan, X., Guo, J., Lan, Y., Xu, J., Cheng, X.: A probabilistic model for bursty topic discovery in microblogs. In: AAAI (2015)
20. Yin, H., Cui, B., Lu, H., Huang, Y.: A unified model for stable and temporal topic detection from social media data. In: ICDE (2013)
21. Huang, W., Chen, W., Zhang, L., Wang, T.: An efficient online event detection method for microblogs via user modeling. In: Li, F., Shim, K., Zheng, K., Liu, G. (eds.) APWeb 2016. LNCS, vol. 9931, pp. 329–341. Springer, Heidelberg (2016). doi:10.1007/978-3-319-45814-4_27
22. Osborne, M., Petrovic, S., McCreadie, R., Macdonald, C., Ounis, I.: Bieber no more: first story detection using twitter and wikipedia. In: SIGIR Workshop on Time-aware Information Access (2012)
23. Steiner, T., Van Hooland, S., Summers, E.: MJ no more: using concurrent wikipedia edit spikes with social network plausibility checks for breaking news detection. In: WWW (2013)
24. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: a probabilistic taxonomy for text understanding. In: SIGMOD (2012)
25. Wang, F., Wang, Z., Li, Z., Wen, J.-R.: Concept-based short text classification and ranking. In: CIKM (2014)
26. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC/ISWC - 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76298-0_52
27. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: a core of semantic knowledge unifying wordnet and wikipedia. In: WWW (2007)
28. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: SIGMOD (2008)
29. Kuzey, E., Weikum, G.: EVIN: building a knowledge base of events. In: WWW (2014)
30. Leban, G., Fortuna, B., Brank, J., Grobelnik, M.: Event registry: learning about world events from news. In: WWW (2014)
31. Wu, Z., Liang, C., Giles, C.L.: Storybase: towards building a knowledge base for news events. In: ACL-IJCNLP (2015)

32. Trampuš, M., Novak, B.: Internals of an aggregated web news feed. In: Proceedings of 15th Multiconference on Information Society, pp. 431–434 (2012)
33. Leetaru, K., Schrodt, P.A.: GDELT: global data on events, location, and tone, 1979–2012. In: ISA Annual Convention, vol. 2. Citeseer (2013)
34. Xiang, E.W., Cao, B., Hu, D.H., Yang, Q.: Bridging domains using world wide knowledge for transfer learning. TKDE **22**, 770–783 (2010)
35. Faralli, S., Stilo, G., Velardi, P.: Large scale homophily analysis in twitter using a twixonomy. In: IJCAI (2015)
36. Yan, R., Song, Y., Li, C.-T., Zhang, M., Hu, X.: Opportunities or risks to reduce labor in crowdsourcing translation? Characterizing cost versus quality via a pagerank-HITS hybrid model. In: IJCAI (2015)
37. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: ICML (1997)
38. Liu, Y., Loh, H.T., Sun, A.: Imbalanced text classification: a term weighting approach. Expert Syst. Appl. **36**(1), 690–701 (2009)
39. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR **3**, 993–1022 (2003)
40. Wallach, H.M.: Structured topic models for language. Ph.D. thesis, University of Cambridge (2008)
41. Tang, J., Meng, Z., Nguyen, X., Mei, Q., Zhang, M.: Understanding the limiting factors of topic modeling via posterior contraction analysis. In: ICML (2014)
42. Von Luxburg, U.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
43. Petrović, S., Osborne, M., McCreadie, R., Macdonald, C., Ounis, I.: Can twitter replace newswire for breaking news? In: ICWSM (2013)
44. Petrović, S., Osborne, M., Lavrenko, V.: The edinburgh twitter corpus. In: NAACL-HLT (2010)
45. Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., Zheng, X., Xing, E.P., Liu, T.-Y., Ma, W.-Y.: LightLDA: big topic models on modest computer clusters. In: WWW (2015)
46. Röder, M., Both, A., Hinneburg, A.: Exploring the space of topic coherence measures. In: WSDM (2015)