News Event Modeling and Tracking in the Social Web with Ontological Guidance

Viet Ha-Thuc*, Yelena Mejova*, Christopher Harris† and Padmini Srinivasan*
*Computer Science Department, The University of Iowa, IA 52242, USA,
Email: hviet@cs.uiowa.edu
Email: ymejova@cs.uiowa.edu
Email: padmini-srinivasan@uiowa.edu
†Informatics Program, The University of Iowa, IA 52242, USA
Email: christopher-harris@uiowa.edu

***Abstract*—News event modeling and tracking in the social web is the task of discovering which news events individuals in social communities are most interested in, how much discussion these events generate and tracking these discussions over time. The task could provide informative summaries on what has happened in the real world, yield important knowledge on what are the most important events from the crowd's perspective and reveal their temporal evolutionary trends. Latent Dirichlet Allocation (LDA) has been used intensively for modeling and tracking events (or topics) in text streams. However, the event models discovered by this bottom-up approach have limitations such as a lack of semantic correspondence to real world events. Besides, they do not scale well to large datasets. This paper proposes a novel latent Dirichlet framework for event modeling and tracking. Our approach takes into account ontological knowledge on events that exist in the real world to guide the modeling and tracking processes. Therefore, event models extracted from the social web by our approach are always meaningful and semantically match with real world events. Practically, our approach requires only a single scan over the dataset to model and track events and hence scales well with dataset size.**

I. Introduction

It is clear that as humans we are becoming increasingly enmeshed in a virtual and social web. A growing number of social mediums: online forums, twitter, facebook and blogs are engaging millions of individuals globally. Within the political and social realm, a recent Pew project reports that close to a fifth of US Internet users have posted online or used a social networking site for civic or political engagement [1]. Unlike traditional media such as newswire which is created by a small group of journalists, social web media could reflect perspectives of a large community. Such perspectives would be extremely important to many areas from business, education to politics.

This research aims to model and track news events such as the 2008 US presidential election in the social web. More specifically, first we discover which news events are most frequently mentioned in social web. Each event is semantically represented by a multinomial distribution over words which formally models the language people use to discuss the event. For example, when talking about the event 2008 US presidential election, people are likely use words like Obama, McCain, election. To quantify the social popularity of each event, we estimate a measure indicating how much the event is mentioned in the social web data for each time period. Second, given the fact that the nature of an event changes over time, we track event semantic representations and social popularity dynamically to reveal temporal evolutionary trends.

Being able to discover event language models (multinomial distributions), their social popularities and to track them offers a number of benefits. First, the task could provide informative summary views on what has happened in the real world. Second, it could yield important knowledge on what are the most important events from the crowd's perspective at any given point of time. Moreover, it could provide deeper insights on when these events start to get discussed in social media, when these discussions peak, when these discussion decline. The insights extracted from social media could reveal evolutionary trends of these events over time.

Not surprisingly, many popular commercial systems have services to support news event or topic tracking. For instanceGoogle Insight Search[1], Blog Pulse (Trend Search[2] and Blog Scope[3] support tracking news events or users' queries in general in social data (web searches or weblogs). Google Insight computes intensity of the event in a time period by counting the number of web searches containing these terms. Similarly, Trend Search computes the intensity by counting the number of weblog posts containing these terms. However, social data is content-contributor centric and diverse [2]. So, it is likely that the same news events could be discussed (and searched for) in different ways. Figure 1 is an example of the temporal trends generated by Blog Pulse of two queries representing the same event. We see a drastic difference between the intensities of the two queries. Generally, the inability to semantically relate surface words to the underlying concepts (e.g. news events in this study) of the strict query-centric approaches as illustrated above makes it likely to miss many relevant items that use other words to discuss the same events. The framework we propose in this paper avoids such pitfalls.

[1]http://www.google.com/insights/search/
[2]http://blogpulse.com/
[3]http://www.blogscope.net/

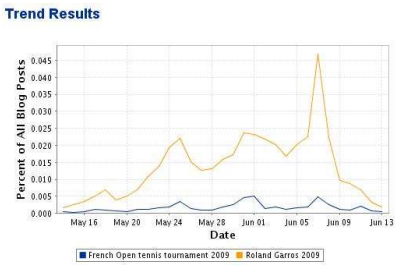

Fig. 1. An Example of Event Tracking

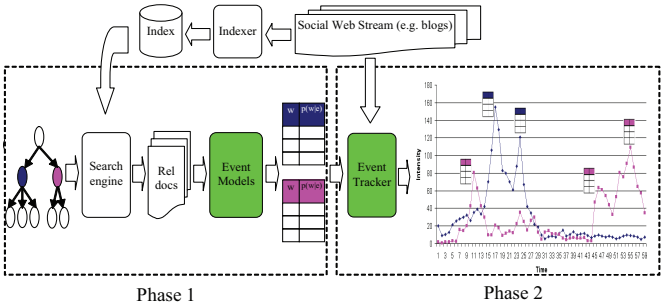

Fig. 2. A latent Dirichlet Framework for news event modeling and tracking

In contrast to query-centric methods, in LDA approach, first one extracts event language models (or topic language models depending on the application) directly from the data, and then one tracks those event models over time [3][4][5]. However, a limitation of the event language models discovered by LDA is that they are synthetic. So, the discovered event models may not correspond to human knowledge. Another consequence is that if we run LDA on two datasets generated in the same period of time there might be little or no correspondence between the two sets of discovered events. This could result in serious problems especially when we seek to integrate models from multiple data sources; a reasonable goal given the multi-faceted nature of the social web. Besides, these approaches do not scale well to large datasets; whereas mass content availability is one of the key strengths of the social web.

Therefore, we propose a novel framework that uses ontological guidance to close the semantic gap between the discovered event language models and events in the real world. The ontological guidance could be a hierarchy of events, where each node is a short title of an event. Given the hierarchy, our system extracts an event language model corresponding to each node in the hierarchy. Compared to the input (event titles), these language models are much richer semantic representations of events. They can capture diverse vocabulary conceptually relevant to the events reflecting how social communities discuss them. After that, the system dynamically tracks event language models to capture semantic evolutions of these events over time. Having dynamic event language models at different time periods, our system computes level of social popularity for these events, measuring how much the community discuss them at each time period. The popularity measure is then used to rank events to discover which events at each abstract level in the hierarchy are most interesting from the crowd's perspective.

II. Proposed Framework

In this study, we propose a novel latent Dirichlet Framework for modeling and tracking events. The overall framework is described in Fig. 2. The social web stream such as a collection of weblogs are crawled, harvested, parsed and then indexed. As mentioned above, our approach takes a hierarchy of events as input. This ontological knowledge is used to guide the modeling and tracking processes so that each discovered event language model is conceptually associated with a node in the hierarchy and therefore to an event in practice.

In the first phase, we compose a query for each event in the hierarchy. Since event titles could be ambiguous and very short, we compose a query for each event by combining the event title and its parent event title (if the event is not at the first level) with the relative weights 2:1. So, this strategy takes into account the hierarchical relationships to automatically enrich queries for events. Then, we submit the queries to the search engine, take the top few documents for each event and assume these to be relevant. We then extract a language model for each event from its corresponding collection of "pseudo-relevant" documents. (Later we refer to this as the "static" language model). The first phase is described in detail in Section III.

In the second phase, given the event language models discovered in the first phase, we track these events in the whole data stream. For this, we first divide the stream into temporal chunks and scan the stream chunk by chunk. For each chunk we refine the static event language models to make the models better fit the social data in the current chunk. The refined language models are then used to compute event social popularity. This measures the relative importance of events from the crowd's perspective during the chunk time span. Tracking these social popularity measures over time, we could understand how the relative importance amongst the events changes and also explore the temporal trend within each event. The second phase is described in detail in Section IV.

III. Phase 1: Generating Event Models

Given training sets containing pseudo-relevant documents (blog posts) for news events in the hierarchy, we will estimate a language model $p(word|event)$ for each of these events. The challenge in estimating the language models from training documents is that these training documents could also contain portions that are non-relevant to the events. For example, a blog post about the event "Lehman Brothers bankruptcy" could also contain background terms or terms relevant to more general events such as "financial crisis". It could also contain terms specific to the local context of the post such as the blogger's proper name. Not removing the general terms could make the language model for the event "Lehman Brothers bankruptcy" overlap heavily and confuse with the language

models for its sibling events such as "Bailout of the US financial system". On the other hand, taking all document-specific terms into account could make the language model for the event over-fit the training set. We address this challenge by proposing a hierarchical latent Dirichlet model for extracting event language models. A flat latent Dirichlet model has also been shown effective in speudo-relevance feedback where a flat list of queries are given in our previous work [6].

A. Model Description

Hierarchical Latent Dirichlet model is a generative model describing the process of generating relevant documents for event in a given hierarchy. Let us denote by W, the number of words in the vocabulary, and by L_t, the level event t in the hierarchy ($L_b = 0$ for the background (root)). Each event t is represented by a multinomial distributionΦ_t , which are sampled from a W-dimensional Dirichlet distribution with hyper-parameters β, denoted by $W - Dir(\beta)$. As any pseudo-relevant document d (for event t) is modelled as a mixture of multinomial distributions of events in the path from the root to t itself and a document-specific language model $t_0(d)$, we denote the corresponding mixture weights by Θ_d. Θ_d is sampled from a Dirichlet distribution with hyper-parameters α. The generative process is formally described as follows:

1. Pick a multinomial distribution Φ_b for the background language model from $W - Dir(\beta)$

2. For each event t in the hierarchy:

2.1 Pick a multinomial distribution Φ_t from $W - Dir(\beta)$

2.2 For each document d relevant to t:

2.2.1 Pick a multinomial $\Phi_{t_o(d)}$ from $W - Dir(\beta)$

2.2.2 Pick a mixing proportion vector Θ_d for elements in set $T = \{background \ldots t, t_0(d)\}$ from $(L_t + 2)$-Dir(α)

2.2.3 For each token in d

2.2.3.1 Pick a multinomial distribution Φ_z in set T from Θ_d

2.2.3.2 Pick a word w from Φ_z

Observe that the scope of background language model (for the root, referred as the top node "event" in our hierarchy) is common for all training documents. The scope of language model Φ_t for event t covers documents in the corresponding sub-tree (i.e. training documents of its and its descendants'). The scope of $\Phi_{t_o(d)}$ includes only document d. Therefore, the background language model will explain words commonly appearing in all training documents of all events (e.g. stop words). Language model Φ_t for each event t generates words relevant to the top level of the sub-tree it represents (too general words are explained by its ascendants, too specific words are explained by language models of its descendants or $\Phi_{t_o(d)}$). In each document d, language model $\Phi_{t_o(d)}$ generates words specific to the context of the document but not relevant to any event from the root to the event t to which the document belongs. All multinomial distributions and mixing proportions in documents are automatically inferred.

B. Inference

Similar to previous work, we also apply Gibbs sampling technique to infer all latent variables (multinomial distributions and mixing proportions in documents) given observed ones (tokens in documents). The algorithm is formally presented in the Figure 3. In Step (1), multinomial distributions (Φ_t) are initialized by maximum likelihood principle from training documents belonging to the corresponding sub-tree, and each $\Phi_{t_o(d)}$ is initialized by maximum likelihood principle from document d. Mixing proportions in all documents are initialized uniformly. In each iteration in Step (2), we sample latent language model generating each token from its posterior. After sampling for all tokens, we update the multinomial distributions and mixing proportions. These sampling and updating sub-steps are repeated until converged. In practice, we set a value for the number of iterations.

1. Initialize all language models $\boldsymbol{\Phi}^{(0)}$ and mixing proportions $\boldsymbol{\theta}^{(0)}$ for all training documents

2. For s = 0 to the desired number of iterations:

2.1 For each token in a training document *d* of an event *t*

Sample the latent index $z^{(s+1)}$ of the language model generating the token in the set P= {all nodes in the path from the root to event *t* }$\cup${t_0(d)} from distribution:

$$p(z|\, w, d) = p(w|\, z)\, p(z|\, d) = \phi_{z,w}^{(s)}\ \theta_{d,z}^{(s)}$$

2.2 Re-estimate all language models $\boldsymbol{\Phi}_t^{(s+1)}$:

$$\Phi_{t,w}^{(s+1)} = p(w|\, t) = \frac{m_{t,w}^{(s+1)} + \beta}{\sum_{w'=1}^{W} (m_{t,w'}^{(s+1)} + \beta)}$$

2.3 For each training document *d* of an event *t*, re-estimate mixing proportion of all language models generating *d*: P= {all nodes in the path from the root to event *t* }$\cup${t_0(d)}

$$\theta_{d,z}^{(s+1)} = p(z|\, d) = \frac{n_{d,z}^{(s+1)} + \alpha}{\sum_{z' \in P} (n_{d,z'}^{(s+1)} + \alpha)} \quad \forall z \in P$$

Fig. 3. Inference Algorithm

IV. Phase 2: Tracking Events

In this phase, given the event language models discovered at the first phase, we track these event models in the whole weblog stream. Specifically, we track the evolution in the language the crowd uses to discuss the events and the evolution in event social popularities. The former likely reflects "semantic drift" of these events. For example, for the US Presidential Election event, around the time of Democratic National Convention, the crowd is likely to talk about Obama, Biden and Democratic Party, while around the time of Republican National Convention, the focus is likely on Republican Party aspect of the event. The later evolution represents "social drift" indicating temporal trend within each event and how the relative importance amongst the events changes over time.

Given the fact that the whole social data stream is often

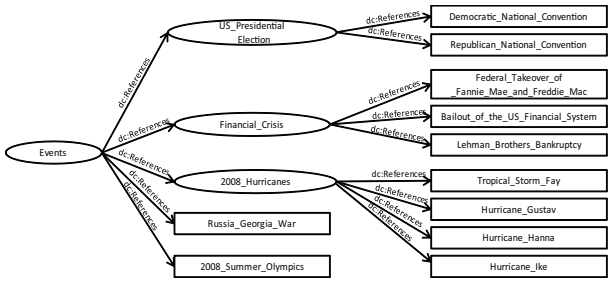

Fig. 4. Event Hierarchy

too large to load into internal memory, we divide the whole blog stream into temporal chunks. We scan through the stream chunk by chunk. So, it takes only one external scan over the whole stream. For each chunk, we use the static event language models as prior knowledge and refine the models to make them better fit data in the current chunk. To extract these dynamic language models, we run an inference algorithm similar the algorithm in Phase 1 (Figure 3). However, in the inference algorithm in Phase 2, static language models are used as initialization in Step (1) and Step (2.2) now becomes as in Equation 1. The numerator and each element in the denominator include two terms. The left term is word count in the current chunk representing the likelihood, while the right terms represents the prior. The "scaling" parameter μ indicates the relative importance between them.

$$\Phi_{z,w}^{(s+1)} = \frac{m_{z,w}^{(s+1)} + \mu * \Phi_{z,w}^{static}}{\sum_{w'=1}^{W}(m_{z,w'}^{(s+1)} + \mu * \Phi_{z,w'}^{static})} \quad (1)$$

After running the inference algorithm, we compute the social popularity of each event at each time point p with window size L as in Equation (5). The measure indicates how much event e_t is mentioned in the subset $C[p, p+L]$ of weblog posts written in the period $[p, p+L]$. Temporal tracking the measure over chunks, we could understand social evolutionary trend of the corresponding event.

$$\begin{aligned} Intensity(e_t, p) &= p(e_t|C[p, p+L]) && (2)\\ &= \frac{\sum_{d\in C[p,p+L]} p(e_t|d)p(d)}{p(C[p, p+L])} && (3)\\ &\approx \sum_{d\in C[p,p+L]} p(e_t|d) && (4)\\ &\approx \sum_{d\in C[p,p+L]} \theta_{d,e_t} && (5) \end{aligned}$$

V. Experimental Study

The data we use for our experiments is provided by weblog indexing service Spinn3r[4]. It includes 60 million postings spanning August and September 2008. We indexed this blog dataset using Lucene[5].

[4]http://spinn3r.com

[5]http://lucene.apache.org/

Figure 4 shows the event hierarchy used in our experiments. The procedure that we used to build the hierarchy is as follows. We used the English Wikipedia[6] to get a list of events during the time span, then we picked the most prominent ones and organized them into a hierarchical structure. The parent-child relationship is defined as follows. An event A subsumes a sub-event B if (i) the time span of A covers the time span of B, and (ii) a document that is relevant to B is also relevant to A.

A. Extracting Static Event Models

In this first experiment, we extract event language models, using our approach and using LDA as a baseline. Tables 1 (upper and lower parts) shows the top probable terms of the event language models for events at the first level in the hierarchy extracted by LDA and by our approach respectively. For the baseline, we run LDA on a random subset of 10,000 documents[7]. To make it comparable, the number of events for the two approaches are set to be the same (*K*=5). One may observe that none of the language models extracted by LDA are unequivocally associated with the news events that happened in that time span. On the other hand, the event language models discovered by our approach are clearly meaningful and conceptually associated with the news events at the first level of the hierarchy.

Table 2 (upper and lower parts) show the language models produced by LDA and by our approach for the sub-events in the domain of "Financial Crisis". For the baseline, we run LDA on 10000 documents belonging to the domain [8] with the same value of *K* to make the results comparable. Notice that the results extracted by LDA rank very general terms like *said* and *know*, as well as the terms belonging to the super event such as *crisis*, *financial* very high. The later terms are still identified as relevant to the sub-events but in reality these are not that important as they do not help to distinguish between sub-events of the same domain. Our approach, on the other hand, extracts sub-event models that strong match with real sub-events in reality. Our experimental results on the other specific domains reveal similar findings. Due to the space limit, the results are not shown here.

To summarize, the findings in this section are two-fold. First, although LDA has been shown to be able to discover meaningful topic language models in other domains [7][8][9][4][10], it fails to discover meaningful event language models in social media such as blogs. We hypothesize that the reason may be because news events appear sparsely in the social data. Only portions of blog posts are about prominent news events. These events are often mentioned in combination with other topics (e.g some personal story) in blog posts. Second, this section confirms that our approach is able to rule out terms in training documents that are on other topics. Our approach with some simple ontological guidance extracts event

[6]http://en.wikipedia.org

[7]10,000 documents is reasonably sufficient compared to previous work on LDA

[8]These documents are the top ranked ones in the document set returned by the search engine when we submit the domain title as query

Event 1		Event 2		Event 3		Event 4		Event 5	
w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$
said	0.006	don't	0.012	day	0.009	new	0.013	other	0.006
span	0.006	know	0.012	time	0.008	video	0.006	online	0.0048
new	0.006	think	0.01	great	0.006	please	0.005	work	0.0042
style	0.006	really	0.009	week	0.006	free	0.005	buy	0.0039
top	0.005	people	0.008	home	0.006	power	0.005	time	0.0037
news	0.005	want	0.007	night	0.006	page	0.004	need	0.0037
US Presidential Election		Financial Crisis		2008 Summer Olympics		Russia-Georgia War		2008 Hurricanes, Tropical Storms	
w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$
election	0.058	financial	0.085	olympics	0.088	georgia	0.071	hurricane	0.08988
presidential	0.048	crisis	0.070	summer	0.050	russia	0.063	storm	0.08785
obama	0.022	bank	0.021	beijing	0.042	war	0.049	tropical	0.05444
vote	0.021	market	0.013	ceremony	0.016	russian	0.041	atlantic	0.01594
candidate	0.015	economy	0.012	gold	0.016	georgian	0.028	season	0.01467
mccain	0.013	economics	0.008	game	0.015	south	0.025	ocean	0.01368

TABLE I
LANGUAGE MODELS FOR TOP NEWS EVENTS DISCOVERED BY LDA (UPPER) AND THE PROPOSED APPROACH (LOWER)

Event 1		Event 2		Event 3	
w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$
financial	0.018	said	0.006	mccain	0.019
government	0.009	new	0.006	obama	0.014
market	0.008	world	0.005	crisis	0.01
banks	0.008	financial	0.005	said	0.008
crisis	0.008	year	0.004	john	0.007
money	0.007	percent	0.004	people	0.006
Federal Takeover of Fannie Mae Freddie Mac		Lehman Brothers Bankruptcy		Bailout of the US Financial System	
w	$p(w\|z)$	w	$p(w\|z)$	w	$p(w\|z)$
fannie	0.061	lehman	0.11085	bailout	0.05834
freddie	0.058	bankruptcy	0.05339	system	0.03265
mae	0.047	brother	0.04602	financial	0.02706
mac	0.046	file	0.0328	plan	0.01484
mortgage	0.03	bank	0.01981	republican	0.01301
federal	0.022	investment	0.01353	congress	0.01225

TABLE II
LANGUAGE MODELS FOR SUB-EVENTS IN THE FINANCIAL CRISIS DOMAIN DISCOVERED BY LDA (UPPER) AND THE PROPOSED APPROACHED (LOWER)

static		Aug 23, 2008		Aug 27, 2008		Sept 3, 2008	
1	election	3	obama	3	obama	6	mccain
2	presidential	*	biden	*	democrat	3	obama
3	obama	*	democrat	6	mccain	*	palin
4	vote	*	senator	*	barack	*	republican
5	candidate	*	barack	*	clinton	*	john
6	mccain	6	mccain	4	vote	8	democrat

TABLE III
LANGUAGE MODELS DISCOVERED DURING VARIOUS STAGES OF TEMPORAL TRACKING FOR EVENT US PRESIDENTIAL ELECTION

language models conceptually matching in with events in the real-world at different abstract levels.

B. Tracking Dynamic Event Models

One of the strengths of our approach is the ability to adapt event language models over time. Table 3 shows several versions of the discovered language model for *US Presidential Election* event. The numbers indicate ranks of words in the static models. Words marked with an asterisk are those that appear in the top 6 positions of that version's language model and not in the static model. On August 23 our model gained terms *senator*, *joe*, *biden*, and *run*, which is the day the Obama campaign announced that Senator Biden would become Barack Obama's running mate. A similar pattern occurs in the August 27 version of the language model when Sarah Palin became the official running mate of John McCain.

C. Tracking Event Social Popularities

In this section we show the effectiveness of our approach for tracking event social popularities described in Section IV. We compare our approach to a baseline method that follows ideas used by commercial systems mentioned earlier. Specifically, the baseline computes intensity of an event over a sliding window of time. It does so by counting the number of blog posts in the whole corpus relevant to the event normalized by the total number of blog posts in the window. The number of relevant blog posts is determined by the search engine when we submit the event title as a query.

Evaluation is a challenge as, to the best of our knowledge, gold standard judgments on event social popularity tracking are not available. Here, we use results returned by Google Insight as a point of reference (not as a gold standard). Note that the results returned by Google Insight are very sensitive to how one describes the events as discussed earlier in Section I. To increase reliability of this as a reference, given an event, we manually try many different descriptions (e.g. "Russia Georgia war" and "South Ossetia war") and take the one with the highest frequency. Note that the intensity measures used by the different approaches are computed using different formula; so the absolute values are not comparable. However, the temporal curves of the same event and relative order amongst events determined by the approaches are comparable. Finally, we use human knowledge made with the help of news sources to evaluate the sensibility of the results.

Figure 5 shows the development of the events at the first level of our event hierarchy (our experimental results at the lower level reveal similar findings. Due to the space limit, the results are not shown here). We can see that the baseline approach fails to capture the peaks in discussion that our novel latent Dirichlet framework clearly shows. In our results,

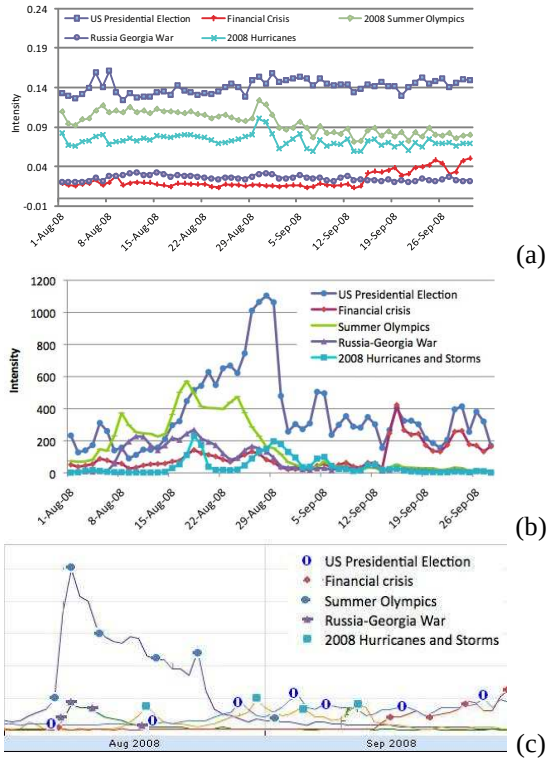

Fig. 5. Temporal intensity trends of top news events discovered by the baseline (a), by the proposed model (b), by Google Insight with manually selected event descriptions (c)

generally each event social popularity curve peaks in intensity at time when the corresponding event in the real world reaches some important milestones. For example, examination of Figure 5 shows a peak in the Beijing Olympics event curve on August 18, the day after closely-watched Michael Phelps won his record eighth gold medal. The announcements of U.S. Presidential running mates by the two major political parties and their National Conventions occurred from August 23 through September 4. A marked increase in social popularity of the US Presidential Election event occurred during this time. Likewise, the *Russia-Georgia Conflict* event is nearly immeasurable until August 8, the day following the launch of a military attack in South Ossetia (wikipedia.org).

Comparing Google Insight with manual query selection to our approach on the US Presidential Election, we see that our results peak strongly when the national conventions occurred at the end of August, whereas the Google almost does not capture this important point. Another notable difference is on Summer Olympics event. This event is very dominant compared to the other events in Google Insight results, but much less dominant in our results (though this event still has similar temporal trends in the two cases). This is perhaps because of the difference in nature between the two data sources.

VI. Conclusion

In this paper, we propose a novel latent Dirichlet framework that uses ontological guidance (event hierarchy) to model and track news events in social web streams. The key advantages of the proposed approach are as follows. First, by using guidance in the modeling and tracking processes, the discovered event language models are always well-defined and they semantically match news events in the real world. Second, the framework takes advantage of relationships defined in the hierarchy in retrieving pseudo-relevant documents and extracting event language models. Third, the framework is robust to noise in the pseudo-relevant documents by automatically ruling out portions that are either too general or too specific. Finally, our system takes only one external scan over the stream, so it scales well to practical social web collections.

Our experiments confirm that with simple ontological knowledge (event titles in the hierarchy), our framework is able to discover event language models that are much more meaningful than the ones discovered by LDA. In terms of social popularity tracking, the temporal trends inferred by our model are more accurate than trends inferred by the query-centric approach. Moreover, the semantic drifts of event language models provide valid insights into the evolution of the events. In future work, we plan to apply our framework for modeling and tracking news events from multiple social web streams and in several languages. Due to the ability of the framework to conceptually connect event language models to events or topics in a formal structured knowledge, language models discovered from different streams or languages about the same event will be naturally aligned. On the other hand, the framework also allows these language models to drift in a way that naturally fits each particular stream. In this way we can mine different perspectives from multiple communities on the same events or topics.

References

[1] A. Smith, K. Schlozman, S. Verba, and H. Brady, "The internet and civic engagement," 2009.
[2] M. Hurst and S. Dumais, "What should blog search look like?" in *Proceedings of the 2008 ACM Workshop on Search in social media*, 2008.
[3] C. Chemudugunta, P. Smyth, and M. Steyvers, "Modeling general and specific aspects of documents with a probabilistics topic model," in *Proceedings of the 2008 IEEE International Conference on Data Mining*, 2006, pp. 241–248.
[4] Q. Mei and C. Zhai, "Discovering evolutionary theme patterns from text: an exploration of temporal text mining," in *Proceedings of KDD 05*. ACM Press, 2005, pp. 198–207.
[5] X. Wang, C. Zhai, X. Hu, and R. Sproat, "Mining correlated bursty topic patterns from coordinated text streams," in *Proceedings of KDD 07*, 2007.
[6] V. Ha-thuc and P. Srinivasan, "A latent dirichlet framework for relevance modeling (lncs)," 2009.
[7] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JOURNAL OF MACHINE LEARNING RESEARCH*, vol. 3, pp. 993–1022, 2003.
[8] E. Erosheva, S. Fienberg, and J. Lafferty, "Mixed membership models of scientific publications," in *Proceedings of the National Academy of Sciences*, 2004.
[9] T. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, pp. 5228–5235, 2004.
[10] M. Steyvers and T. Griffiths, "Probabilistic topic models," 2006.