A Semantic Graph-Based Approach for Mining Common Topics from Multiple Asynchronous Text Streams

Long Chen[†],Joemon M Jose[†], Haitao Yu[‡], Fajie Yuan[†]
[†]University of Glasgow, UK [‡]University of Tsukuba, Japan
long.chen@glasgow.ac.uk, joemon.jose@glasgow.ac.uk,
yuhaitao@slis.tsukuba.ac.jp, f.yuan.1@research.gla.ac.uk

ABSTRACT

In the age of Web 2.0, a substantial amount of unstructured content are distributed through multiple text streams in an asynchronous fashion, which makes it increasingly difficult to glean and distill useful information. An effective way to explore the information in text streams is topic modelling, which can further facilitate other applications such as search, information browsing, and pattern mining. In this paper, we propose a semantic graph based topic modelling approach for structuring asynchronous text streams. Our model integrates topic mining and time synchronization, two core modules for addressing the problem, into a unified model. Specifically, for handling the lexical gap issues, we use global semantic graphs of each timestamp for capturing the hidden interaction among entities from all the text streams. For dealing with the sources asynchronism problem, local semantic graphs are employed to discover similar topics of different entities that can be potentially separated by time gaps. Our experiment on two real-world datasets shows that the proposed model significantly outperforms the existing ones.

General Terms

Algorithm, Experimentation

Keywords

Knowledge Repository; Topic Modelling; Language Modelling

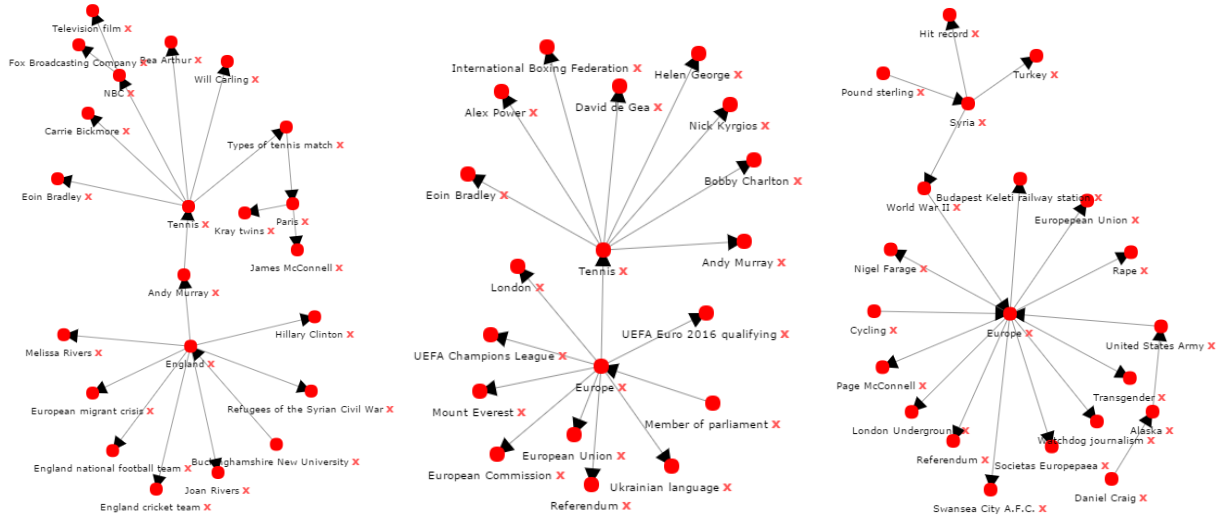
1. INTRODUCTION

With advent of Web 2.0 and increased connectivity, various kind of text streams are published online such as news feeds and weblog articles. One interesting characteristic of such text streams is that there are usually intensive coverage of some common topics within a certain time frame. For example, when Ukrainian crisis happened in 2014, all news articles tend to have intensive coverage of the event. Similarly, when a new research direction is opened up in a research field, many researchers tend to produce publication towards the same direction for a certain time period. Mining topic trends from multiple streams can facilitate tasks such as search, mining, and documents summarization.

Intuitively, mining topics from multiple sources can find more meaningful and comprehensive topics than that of a single source. However, this is often a difficult task, since each stream always owns a unique vocabulary, which would lead to a lexical gap between the words of different streams. In addition, the asynchronous communication of different streams may cause topic mismatch. To address the above limitations, *Cross-Collection Model* [12, 30] is proposed by assuming a shared time distribution of words across multiple streams. While generally useful, the relations between any two words of different streams can only be indirectly inferred by a multinomial words distribution (also known as *unigram language model*) over each timestamp. However, words in different streams about the same topics may not necessarily share a similar frequency distribution over time. Furthermore, a simple word-level analysis model cannot handle cases where entities (e.g., *people, places, and concepts*) are expressed by multi-word phrases. To go beyond the mere word-level analysis, this paper proposes a topic modeling framework on the basis of semantic graphs which makes use of an external resource (namely DBpedia) as the background knowledge to bridge the lexical gap and address the topic synchronous problem in a principled way.

As a simple illustration, Figure 1 displays three pieces of local semantic graphs produced by our proposed knowledge-rich approach (cf. Section 4.2.1, the infrequent entity threshold is set as 4 to reduce the graph size) from NEWSIR datasets (cf. Section 5.1, where we use day as the timestamp since it is the finest granularity available in this dataset). One can easily see that the news reports was focused on "Europe", "England", and "Tennis" , as they are always the central entities (i.e., *concepts* in *DBpedia*) of the graphs. In addition, it is clear that the first two semantic graphs share a closer resemblance than that between the first and the third one, which suggests that the inference of a timestamp's latent topics could benefit from examining its similar timestamps based on their individual local semantic graphs. In other words, if two timestamps have a large degree of overlap in terms of their semantic entities and relations, they probably bear a close topical resemblance to each other as well. Therefore, we construct local semantic graphs for each timestamp in the archive with the hope to utilize their semantic similarities to overcome the asynchronous problem.

WWW 2017, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052630



(a) NEWSIR on 01/09/2015 (b) NEWSIR on 02/09/2015 (c) NEWSIR on 03/09/2015

Figure 1: Three pieces of local semantic graphs generated from NewsIR datasets

To bridge the lexical gap among multiple streams, global semantic graphs are also proposed, which is built with the entire documents collection. The intuition is that the rich semantic connections among entities could be exploited to help inferring the latent topics of documents that derived from varying stream. Therefore, we would like to learn the inter-relationships between entities in the global semantic graph, which would allow effective information sharing among all streams.

Given the observation above, a novel topic modelling framework is proposed, namely, Semantic Graph based Mixture Model (**SGMM**), which can seamlessly incorporate entities and relations from the semantic graphs into topic modelling. Experiments on two real-world datasets show that the semantic-based approach outperforms existing ones significantly.

The rest of this paper is organized as follows. We firstly introduce the related work in Section 2. Section 3 formally defines the problem of topic modelling with semantic graph. Section 4 systematically presents the proposed *SGMM* framework. The experimental results of topic modeling are reported in Section 5. Finally, we present our conclusion and future work in Section 6.

2. RELATED WORK

2.1 Topic Model with Network Analysis

The techniques of topic modelling, such as PLSA [11] and LDA [31], provide an elegant mathematical way to analyze large volumes of unlabelled text. Recently, a large number of studies, such as Author-Topic Model (ATM) [25] and Contextual Focused Topic Model (CFTM) [7], try to integrate some information of network structure with topic modelling, but they mostly focus on homogeneous networks rather than heterogeneous networks. Entity-Topic Model (ETM) [15] combined LDA with entity-document relations, which is somewhat similar to our idea. However they assume that an edge (entity-document) created in exactly the same way as a word, whereas our approach directly takes into account several types of relations (entity-document and entity-entity relations) through regularized propagation.

2.2 Knowledge Rich Approaches

The recent advances in knowledge-rich approaches (e.g., DBpedia[1] and Knowledge Graph[2]) provide new techniques to gain insight into the semantic structure of a dataset. While enormous success has been made in several NLP tasks such as document similarity [23], topic labelling [14], and question answering [6], the feasibility and effectiveness of such knowledge-rich approaches in topic modelling and tracking are mostly unknown. Hulpus et al. [14] reported a framework that extracts sub-graphs from DBpedia to label the topics obtained from a topic model. However, they consider topic model and graph labelling as two separate processes, which may result in the loss of rich semantics. On the contrary, our framework discovers the latent topics and semantic network simultaneously, reinforcing the topic model performance with multi-typed relations. In addition, their graph construction process relies on a small set of manually selected DBpedia relations, which does not scale and needs to be tuned each time given a different knowledge repository. Instead, we pruned our semantic graphs by filtering and weighting the edges (see Section 4.3.1). This may look similar to [23], but their work attempts to produce graph-representation of documents for the task of document ranking, while we aim to construct semantic graphs for the task of topic modelling.

2.3 Topic Mining from Multiple Text Streams

Zhai et al. [34] proposed a cross-collection mixture model to detect common topics and local ones respectively, however, their analysis was limited to two static collections. The state-of-the-art approach [12] utilises the *Cross-Collection Model* [34] for mining topics from multiple streams, together with a meme-tracking model that iteratively updates the hyper-parameter that controls the document-topic distribu-

[1]http://wiki.dbpedia.org/
[2]https://developers.google.com/freebase/

tion to capture the temporal dynamics of the topic. In this model, a word belongs to either the local topic or the common topic, the probability of which is drawn from a bernoulli distribution. But it assumes that there is no correspondence between local topics across different sources, nor is there exchanging information between the local topics and the common ones. While in real-world scenarios, information from multiple streams constantly interacts with each other as the topic evolves. To address this problem, Ghosh et al. [9] proposed *Source LDA* to detect topics from multiple sources with the aim to exploit the source interactions, which is somewhat similar to our idea. However, there are stark differences between their work and ours: First, their model assumes that there is no order for the documents in the collection, hence the temporal dynamics of each source is completely ignored. Secondly, since the same LDA model is simply applied over multiple streams, they didn't exploit external resources to facilitate the communication among multiple sources, whereas we use global semantic graph to force multiple streams to interact with each other.

3. PRELIMINARIES

In this section, we formally introduce several concepts and notations.

Definition 1 (Text Stream): A ***text stream*** S is a sequence of $\{d_{1,s}, ..., d_{N,s}\}$, in which $d_{i,s}$ is a sequence of words $\{w_1, ..., w_{|d|}\}$. Each document correspond to a unique timestamp t, and each **timestamp** is composed of a documents collection over multiple text streams $\{S_1, ..., S_S\}$. Following a common simplification used in most work in information retrieval [11], we consider each document as a bag of words, and use $n(d_{i,s}, w)$ to denote the number of occurrence of word w at document $d_{i,s}$.

Definition 2 (Entity): An **entity** e in our system can be either an instance or a concept in DBpedia URI [3]. The former are concrete entries of DBpedia [4], while the latter are the classes found within the DBpedia Ontology (i.e., the types of instances such as people, places, organizations).

Definition 3(Semantic Graph): A **semantic graph** G consists of V and E, the former is a set of entities and the latter is a set of edges representing the relations between the entities. For instance, an edge $< u, v >$ is a binary directed relation from entity u to entity v, where we use $w(u, v)$ to denote the weight of $< u, v >$. We define the semantic graph built from the documents of $\{t_0, ..., t_T\}$ the *global semantic graph*, and those built from the documents of a single timestamp t the *local semantic graphs*.

Now we can formulate our task of topic mining as follows. Given a set of text streams $\{S_1, ..., S_S\}$ with a set of semantic graphs $\{G_1, ..., G_L\}$, in which $G_l = (V_l, E_l)$, we would like to extract K common topics $Z = \{z_1, z_2, \ldots, z_K\}$ from $\{S_1, ..., S_S\}$, where z_k is a probability distribution of words, and the probability of a word w appeared in topic z_k is denoted as $P(w|z_k)$, the probability of z_k over a timestamp t is denoted as $P(z_k|t)$. Notice that semantic information of a topic is encoded by the conditional distribution $P(w|z_k)$ and $P(z_k|t)$. What *SGMM* essentially does is to group similar or related documents of different timestamps and different streams into semantic clusters considering not only their textual similarities, but also the hidden entity relations in semantic graphs.

[3]http://dbpedia.org/page/Uniform_Resource_Identifier

[4]e.g. *dbpedia* : *Barrack_Obama*

4. TOPIC MODELS

In this section, we propose a propagation algorithm to combine semantic graphs with the textual information of multiple streams for topic modelling, namely **Biased Propagation**. The goal of this algorithm is to estimate the probabilities of topics for documents as well as other associated entities, in order to improve the performance of topic modelling.

4.1 Simple Mixture Model

A naive solution for common topic mining is to treat the multiple streams as a single stream and perform topic modelling. We now present a simple mixture model for topic mining from multiple streams. In simple mixture model (**SMM**) [28] , an unobserved topic variable $z_k \in \{z_1, ..., z_K\}$ is inferred from the occurrences of different words $w_j \in \{w_1, ..., w_M\}$ in documents $d_i \in \{d_1, ..., d_N\}$ at a particular timestamp t. The joint probability of observed triplets (s, d, w) can be expressed as

$$P(S_s, d_{i,s}, w_j) = P(d_{i,s}) \sum_{k=1}^{K} P(w_j|z_k)P(z_k|t) \quad (1)$$

$$P(z_k|t) = \sum_{i=1}^{N}\sum_{k=1}^{K} P(z_k|d_{i,s})P(d_{i,s}|t) = \sum_{i=1}^{N}\sum_{k=1}^{K} \frac{P(z_k|d_{i,s})P(t|d_{i,s})P(d_{i,s})}{P(t)} \quad (2)$$

where $P(w_j|z_k)$ is the probability of word w_j occurring in topic z_k, $P(z_k|t)$ is the probability of topic z_k for timestamp t over the documents derived by all the streams S. Then, $P(z_k|d_{i,s})$ and $P(w_j|z_k)$ can be estimated by maximizing the log likelihood over all the streams S as follows:

$$L(S) = \sum_{s=1}^{S}\sum_{i=1}^{N}\sum_{j=1}^{M} n(d_{i,s}, w_j) \log \sum_{k=1}^{K} P(w_j|z_k)P(z_k|t) \quad (3)$$

$$P(z_k|t) = \sum_{i=1}^{N}\sum_{k=1}^{K} \frac{P(z_k|d_{i,s})P(t|d_{i,s})n(d_{i,s}, w_j)}{\sum_{s=1}^{S}\sum_{i'=1}^{N}\sum_{j'=1}^{M} n(d_{i',s}, w_{j'})} \quad (4)$$

Since a given document has and only has one timestamp, the value of $P(t|d_{i,s})$ is either 0 or 1. So the problem now lies in how to learn $P(z_k|d_{i,s})$ and $P(w_j|z_k)$, which can be estimated in a similar fashion as PLSA (Probabilistic Latent Semantic Analysis [11]) using EM algorithm (cf. Section 4.3.2). In *SMM*, a word is considered as generated from the shared time distribution of words across all streams, and the asynchronism among multiple streams is alleviated by exploiting a window model [28]. However, it is not adequate, as indicated in [30], for modeling text from different streams for two reasons: (1) The structure of collections is completely ignored. Consequently, the extracted common topics might only represent some, but not all collections; (2) It is hard to determine which topic correspond to the common information across streams and which correspond to specific information to a particular stream.

4.2 Cross-Collection Model

To overcome the above limitations, Cross-Collection Mixture Model (**CCMM**) [12, 34] is proposed, which can explicitly distinguish common topics that characterize common information across all streams from local topics that characterize stream-specific information. In this model, λ_B controls the weight of background language model for structuring asynchronous streams. In addition, one needs to decide whether to use the common topic model or the stream-specific topic model for a given topic, which is controlled by the trade-off parameter λ_C. Formally speaking, we now consider K common topics as well as a potentially different set of K local topics for each stream. The word distribution in document d (from stream S_s) is now stream-specific, which involves the unigram language model (θ_B), K common topic models ($\theta_1, ..., \theta_K$), and K local topic models ($\theta_{1,s}, ..., \theta_{K,s}$). The joint probability of an observed triplet (s, d, w) can thus be represented as

$$P_{cross}(S_s, d_{i,s}, w_j) = \lambda_B P(d_{i,s}) \sum_{k=1}^{K} [\lambda_C P(w_j|z_k) P(z_k|t) + (1-\lambda_C) P(w_j|z_{k,s}) P(z_{k,s}|t)] + (1-\lambda_B) P(w_j|\theta_B) \quad (5)$$

The model parameters can be estimated by maximizing the log likelihood of all the streams S as

$$L_{cross}(S) = \sum_{s=1}^{S} \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_{i,s}, w_j) \log[\lambda_B \sum_{k=1}^{K} [\lambda_C P(w_j|z_k) \cdot P(z_k|t) + (1-\lambda_C) P(w_j|z_{k,s}) P(z_{k,s}|t)] + (1-\lambda_B) P(w|\theta_B)] \quad (6)$$

where $P(z_{k,s}|t)$ and $P(z_k|t)$ are estimated in the same way as *SMM* using the Equation 4. As mentioned before, *CCMM* has no constraint on the parameters $\psi_k = P(z_k|t)$, the number of which grows linearly with the data. Therefore, the model tends to overfit the data.

4.3 Semantic Graph based Mixture Model

CCMM uses coarser granularity of the timestamps such that the asynchronism among streams can be smoothed over via a background language model θ_B. This is apparently dissatisfactory as it may cause unbearable loss in the temporal information of common topics and different topics would be inevitably intertwined. In this section, we propose a biased propagation algorithm to incorporate the semantic graph with the textual information for topic modelling, so as to estimate the probabilities of topics for each timestamp as well as other associated entities across streams, and consequently addressing asynchronism among multiple streams.

Specifically, given the topic probability of a document $P(z_k|d_{i,s})$, the topic probability of an entity can be calculated by:

$$P_{sec}(z_k|e) = \frac{1}{2}\Big(\sum_{d_{i,s} \in D_e} P(z_k|d_{i,s}) P(d_{i,s}|e) + \sum_{e_j \in C_e} P(z_k|e_j) P(e_j|e)\Big) \quad (7)$$

$$= \frac{1}{2}\Big(\sum_{d_{i,s} \in D_e} \frac{P(z_k|d_{i,s})}{|D_e|} + \sum_{e_j \in C_e} P(z_k|e_j) P(e_j|e)\Big) \quad (8)$$

where D_e is a set of documents in the current timestamp which contain the entity e, C_e is a set of entities which are connected to entity e through semantic graph. $P(z_k|e_j)$ is the topic probability of entity e_j, which is estimated with a similar manner as $P(z_k|d_{i,s})$ by using the EM algorithm (see Section 4.3.2). $P(e_j|e)$ is the highest weight between entity e_j and e (see Section 4.3.1). The underlying intuition behind the above equation is that the topic distribution of an entity is determined by the average topic distribution of connected documents as well as the connected entities of semantic graph. On the other hand, the topic distributions could be propagated from entities to documents, so as to reinforce the topic distribution of time. Thus we propose the following topic-documents propagation based on semantic graph:

$$P_E(z_k|d_{i,s}) = \xi P(z_k|d_{i,s}) + (1-\xi) \sum_{e \in E} \frac{P(z_k|e)}{|E|} \quad (9)$$

$$P_{sec}(z_k|d_{i,s}) = \lambda P_{E_g}(z_k|d_{i,s}) + (1-\lambda) P_{E_l}(z_k|d_{i,s}) \quad (10)$$

$$P_{sec}(S_s, d_{i,s}, w_j) = P(d_{i,s}) \sum_{k=1}^{K} [\lambda_C P(w_j|z_k) P_{sec}(z_k|t) + (1-\lambda_C) P(w_j|z_{k,s}) P_{sec}(z_{k,s}|t)] \quad (11)$$

$$P_{sec}(z_k|t) = \sum_{s=1}^{S} \sum_{i=1}^{N} \sum_{k=1}^{K} P_{sec}(z_k|d_{i,s}) P(d_{i,s}|t) \quad (12)$$

where E denotes the set of entities of document $d_{i,s}$, ξ is the biased parameter to strike the balance between inherent topic distribution $P(z_k|d_{i,s})$ and entity topic distribution $P(z_k|e)$. If $\xi = 1$, the topics of the document retain the original ones. If $\xi = 0$, the topics of the document are determined by the entity topic distribution. $P_{E_g}(z_k|d_{i,s})$ is propagated through the global semantic graph, and $P_{E_l}(z_k|d_{i,s})$ is propagated through the local semantic graph. The log-likelihood of *SGMM* can then be given as

$$L_{sec}(S) = \sum_{s=1}^{S} \sum_{i=1}^{N} \sum_{j=1}^{M} n(d_{i,s}, w_j) \log \sum_{k=1}^{K} [\lambda_C P(w_j|z_k) P_{sec}(z_k|t) + (1-\lambda_C) P(w_j|z_{k,s}) P_{sec}(z_{k,s}|t)] \quad (13)$$

4.3.1 Mapping Documents into Semantic Graphs

When computing $P(e_j|e_u)$ in the above *SGMM*, the method of [23] is adopted to construct the semantic graph. We start with a set of input entities C, which is found by using the off-the-shelf entity recognition tool DBpedia Spotlight[5].

We then create a directed graph G as follows: 1) we define the set of entities V of G to be made up of all input entities, i.e., we set $V := C$; 2) we connect the entities in V based on the directed paths found between them in DBpedia. Specifically, the set of entities in V are expanded into a graph by conducting a depth-first search along the DBpedia graph and adding all the visited relations and entities, to a certain limit. So the finally constructed semantic graph consists of all the "seed" entities identified from the documents together with all the edges found along the paths up to maximal length L that connect them. In this work, we set $L = 2$, as we find that the model with $L > 2$ tends to produce very large graphs and introduce lots of noise.

[5]https://github.com/dbpedia-spotlight/dbpedia-spotlight

So far, we simply traverse a set of input entities from DBpedia graph. However, recall that entities can be divided as instance entities and concept entities, the latter contains semantic relations at different levels which may not be equally informative. For example, two entities can be connected through two *Predicate Types* [6] of **rdf:type foaf:person** and **dbpprop:birthPlace**, but the former is less informative since it can apply to a large number of entities (i.e., all people in DBpedia). We can use real-valued weights to describe the degree of correlation between entities in the graph, and the core idea underlying our weighting scheme is to reward those edges that are most specific to the entities connected by them. Therefore we define the weighting function as

$$W = -\log(P(W_{Pred})) \tag{14}$$

where W is the weight of an edge, $P(W_{Pred})$ is the probability that the predicate W_{Pred} (such as **rdf:type**) is describing the specific semantic relation. This measure is based on the hypothesis that specificity is a good estimator for relevance. We can compute the document frequency for each type of predicates, as we have the whole DBpedia database available and are able to query for all possible realizations of the variable X_{Pred}. $P(W_{Pred})$ is then defined in the same way as the tf-idf [20] representation of W_{Pred}. There are often multiple relations between two entities, so the relation with the highest weight will be selected as the final edge.

4.3.2 Model Fitting with the EM Algorithm

To estimate $P(w|z_k)$ and $P(z_k|t)$ in the *SGMM*, we use the Expectation Maximization (EM) algorithm, which alternates two steps, E-step and M-step. The unobserved latent variables in our model include $\phi = P(w_j|z_k)$, $\delta = P(z_k|d_{d,s})$, and $\varphi = P(z_k|e_l)$.

In E-step, we calculate the posterior probabilities:

$$P(z_k|d_{i,s}, w_j) = \frac{P(w_j|z_k)P(z_k|d_{i,s})}{\sum_{k'=1}^{K}\sum_{s=1}^{S} P(w_j|z_{k'})P(z_{k'}|d_{i,s})} \tag{15}$$

$$P(z_k|d_{i,s}, e_l) = \frac{P(z_k|e_l)P(e_l|d_{i,s})}{\sum_{k'=1}^{K}\sum_{s=1}^{S} P(z_{k'}|e_l)P(e_l|d_{i,s})} \tag{16}$$

In the M-step, we maximize the expected complete data log-likelihood:

$$Q_D = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{s=1}^{S} n(d_{i,s}, w_j) \sum_{k=1}^{K}\sum_{s=1}^{S} P(z_k|d_{i,s}, w_j) \times \log \sum_{k=1}^{K}\sum_{s=1}^{S} P(w_j|z_k)P(z_k|d_{i,s}) \tag{17}$$

There is a closed-form solution [8] to maximize $L_{sec}(S)$, which are listed in Equation 18, 19, 20 and 21.

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N}\sum_{s=1}^{S} n(t, w_j)P(z_k|d_{i,s}, w_j)}{\sum_{j'=1}^{M}\sum_{i=1}^{N}\sum_{s=1}^{S} n(t, w_{j'})P(z_k|t, w_{j'})} \tag{18}$$

$$P(z_k|d_{i,s}) = \frac{\sum_{j=1}^{M}\sum_{s=1}^{S} n(d_{i,s}, w_j)P(z_k|d_{i,s}, w_j)}{\sum_{j'=1}^{M}\sum_{s=1}^{S} n(d_{i,s}, w_{j'})} \tag{19}$$

$$P(z_{k,s}|d_{i,s}) = \frac{\sum_{j=1}^{M'} n(d_{i,s}, w_j)P(z_{k,s}|d_{i,s}, w_j)}{\sum_{j'=1}^{M'} n(d_{i,s}, w_{j'})} \tag{20}$$

$$P(z_k|e_l) = \frac{\sum_{s=1}^{S} n(t, e_l)P(z_k|d_{i,s}, e_l)}{\sum_{s'=1}^{S} n(t, e_l)} \tag{21}$$

where $n(t, e_l)$ indicates the frequency of entity e_l at timestamp t. Given $P(z_{k,s}|d_{i,s})$, $P_{sec}(z_k|t)$ can be estimated using Equation 4.

5. EXPERIMENTS

5.1 Experimental Setup

We collect streams from two real-world datasets. The first dataset is Twitter of UK dating from September 1, 2015 until September 30, 2015, which serves as the Twitter stream. Since the original dataset is quite large, to speed up the experiments, we follow the common practise as in [12] by randomly sampling Tweets proportional to the total volume of each hour, resulting in 1,218,210 Twitter messages in total. The second dataset, The NewsIR (Signal Media One-Million News Articles Workshop [7]), is a collection of news articles derived from major newswires, such as Reuters, in addition to local news sources and blogs. The articles of the dataset were collected by Moreover Technologies[8] from a variety of news sources (such as Reuters and BBC) for a period of 1 month (1-30 September 2015). We use 5 sources of this collection, namely, Guardian, Mail Online UK, Reuters UK, Yahoo! UK, and Myinforms, each of which serves as an independent stream.

The distribution of each stream of NEWSIR dataset is shown in Figure 2. The datasets' statistic along with their corresponding entities and links are shown in Table 1. We randomly split each of the dataset into a training set, a validation set, and a test set with a ratio 2:1:1. We learned the parameters in the models on the training set, tuned the parameters on the validation set and tested the performance of our model and other baseline models on the test set. The training set and the validation set are also used for tuning parameters in baseline models.

For both datasets, each day is used as a timestamp thus the length of this stream is 30. For preprocessing, all the documents are lowercased and stopwords are removed using a standard list of 418 words. Entities are disambiguated with the off-the-shelf tool DBpedia Spotlight, and we empirically set the confidence value as 0.25. Given the disambiguated entities (cf. 4.3.1), we create local and global entity collections, respectively, for constructing local and global semantic graphs. The creation process of entity collections is organized as a pipeline of filtering operations:

1. The isolated entities, which have no paths with any other entities of the full entity collection in the DBpedia repository, are removed, since they have less power in the topic propagation process.
2. The infrequent entities, which appear in less than five documents when constructing the global entity collection, are discarded.

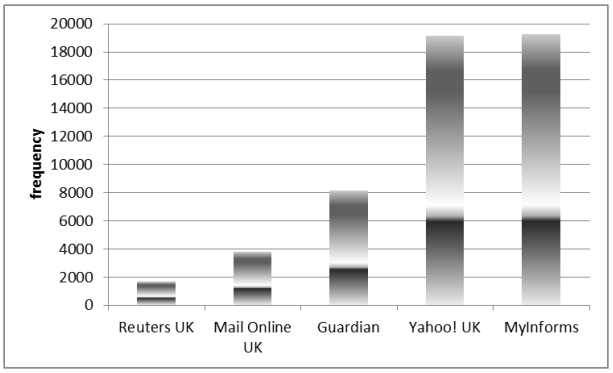
[6] http://mappings.dbpedia.org/server/ontology/classes/

[7] http://research.signalmedia.co/newsir16/signal-dataset.html

[8] http://www.moreover.com/

Table 1: Statistic of Twitter and NEWSIR dataset

	Twitter	NewsIR
# of docs	1,218,210	51,973
# of entities (local)	452,85	249,782
# of entities (global)	473,122	228,502
# of links (local) docs	653,291	486,435
# of links (global) docs	1279,639	874,832



(a) NewsIR

Figure 2: The category distribution of NewsIR datasets

3. Similar to step 2, we discard entities that appear less than three times in the document when constructing the local entity collection.

5.2 Experiments with Topic Modelling

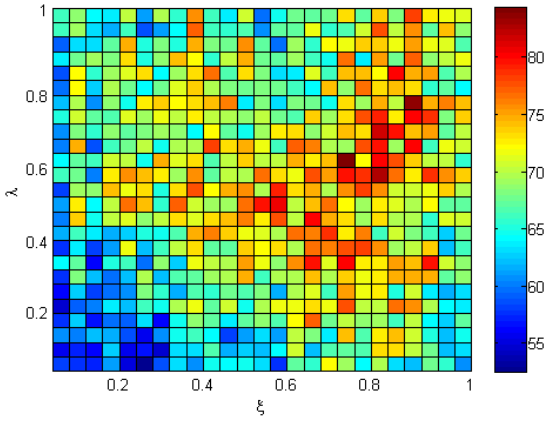


Figure 3: The NMI (%) of SGMM framework with varying parameters λ and ξ.

To demonstrate the effectiveness of our *SGMM* method, we compare it with the following topic modelling techniques:

- **SMM:** The baseline approach [28], which simply merges multiple streams and then apply topic model. (See Section 4.1)
- **CCMM:** The state-of-the-art approach [12,34], which distinguish common topics from local topics and structure asynchronous streams with a background language model. (See Section 4.2)
- **SGMM:** Our proposed Semantic Graph based Mixture Model. (See Section 4.3).

In order to tune the parameters of our proposed topic model, we use the metric of normalized mutual information (NMI), which is a common metric for evaluating the effectiveness of topic modelling [8]. Given two sets of timestamps clusters, C and C', their mutual information is defined as: $MI(C,C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log_2 \frac{p(c_i,c'_j)}{p(c_i)\cdot p(c'_j)}$ [32], where $p(c_i)$ and $p(c'_j)$ are the probabilities that a randomly chosen timestamp belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that a randomly chosen timestamp belongs to the cluster c_i and c'_j at the same time.

Recall that in order to optimize our objective functions, we have two core parameters to estimate, namely $P(z|t)$ and $P(w|z)$. Therefore we will also examine these two parameters in the following experiment.

Parameter Setting: For *SMM* and *CCMM*, we use both titles and mainstory for document clustering with no additional entity information. As reported in [34], the optimal value for λ_B is between 0.9 and 0.95, so we empirically set $\lambda_B = 0.95$, the adjacent window size of background language model is set as 3. For *CCMM*, we use EM algorithm to fit the model with the empirical setting of $\lambda_B = 0.95$ and $\lambda_C = 0.25$, the other parameter settings were set to be identical to those in [34]. Furthermore, it is well known that in general we need to use more topics for larger datasets to achieve the best topic modelling effect. Hence, we tried the mixture topic models with different values of K. To remove bias and variance, 5-fold cross validation is performed on training dataset. As shown in Figure 4 (d), it is clear that all models achieve a relative good results when $K = 200$, which is much larger then the one reported in [29]. One possible reason is that Twitter dataset is noisier than regular dataset, thereby as the number of topics is increased beyond the minimum, overfitting tends to set in, which was also observed in [12].

Following a similar setup as in [12], we set the number of common topics (K) as 200, and equally assign 200 topics into all other streams as local ones. Figure 3 shows how the *SGMM* clustering performance varies with the different parameter values. The essential parameters in the *SGMM* framework are λ and ξ. As mentioned in Section 4.3, ξ controls the relative importance of the inherent textual information against the semantic graph information, and λ controls the balance between the local semantic graph and the global semantic graph. When $\xi = 0$, it is the state-of-the-art *CCMM*. When $\xi = 1$, is is entirely determined by the semantic graph. The results reported in Figure 3 are produced from the training dataset. It can be seen that *SGMM* with global semantic graphs generally performs better than *SGMM* with local semantic graphs, which possibly suggests that the global context is more important than the local context for the purpose of topic modeling. Furthermore, the best performance is achieved when combining these two with the parameter setting: $\lambda = 0.6$ and $\xi = 0.5$.

Table 2: The representative terms generated by SMM, CCMM, and SGMM models. The terms are vertically ranked according to the probability $P(w|z)$.Some of the duplicated topical words are underlined.

	TOPIC 1		TOPIC 2		TOPIC 3		TOPIC 4		TOPIC 5	
SMM	united	fans	against	military	open	women	crisis	migrants	world	cup
	world	football	syria	refugee	murray	davis	refugee	call	tennis	shows
	league	final	russia	russian	andy	kyrgios	europe	hungary	andy	women
	city	champion	strikes	british	defeat	cricket	migrant	help	gea	davis
	club	premier	air	islamic	final	win	eu	plan	de	kyrgios
CCMM	world	scotland	refugees	hungary	video	city	china	global	david	strikes
	cup	win	syrian	thousands	shows	set	update	uk	cameron	uk
	play	final	take	border	photo	show	stocks	brief	syria	china
	wales	opener	britain	welcome	singa	west	open	fed	against	air
	against	italy	europe	help	game	star	oil	shares	russia	oil
SGMM	world	win	refugees	uk	tennis	men	china	minister	corbn	victory
	cup	fiji	david	eu	murray	round	says	bank	jeremy	shadow
	final	against	cameron	crisis	andy	kyrgios	brief	united	labour	leadership
	england	champion	syrian	border	final	player	update	group	party	cabinet
	wallabies	rugby	europe	welcome	open	uk	chief	england	leader	trident

5.3 Results and Analysis

We apply our method on the test dataset with the same setting described in Section 5.2. Following a similar practice in the state-of-the-art [12], we extracted five common topics from the streams with the highest average value of $p(z|t)$. For each topic, ten representative words with the highest probability ($p(w|z)$) were shown in Table 2. One can see that all topics extracted by our method (*SGMM*) are coherent and easy to understand, since global semantic graphs which reward words derived from name entities. For example, "Rugby World Cup 2015", "Shadow Cabinet of Jeremy Corbyn", "Syrian Civil War", "Bank of China", "Andy Murray", etc. All of these topical words accurately depict the most important topics along with their corresponding name entities happened during that time period. Comparing the topics extracted by our method to those by the baseline methods, we can see that our model also provides more discriminative topics. As a contrast, both *SMM* and *CCMM* suffered from the asynchronism in the streams and extracted many duplicated topical words. A possible reason is that documents of asynchronous streams related to different topics may be indexed by the same timestamp, and documents related to the same topic may appear at different timestamps. Our method is able to propagate common topics of different streams by exploiting the local semantic graphs to alleviate this problem.

In order to show the time series of the common topic on NewsIR and Twitter, we adopt the metric proposed in [12], which transforms the counts into a valid distribution by calculating a $P(t|z) = \frac{P(z|t)P(t)}{\sum_{t'}^{T} P(z|t')P(t')}$. This metric is a good indicator about how likely the topic would appear in timestamp t. Figure 4(a), (b), and (c) show the discriminative power of finding common topic "Rugby World Cup 2015" for *SMM*, *CCMM*, and *SGMM* respectively. It is easy to observe that the topic has a major peak on both NewsIR and Twitter around 18 September, shown in all figures, which is consistent with the opening time of "Rugby World Cup 2015". From Figure 4(a), it is interesting to see that the topic first exhibited a peak on NewsIR and exhibited another peak on Twitter days later, since *SMM* only relies on a simple window model to resolve asynchronism. As shown in Figure 4(b), *CCMM* can remove asynchronism even better by smoothing it over the background language model. More importantly, as shown in Figure 4(c), *SGMM* exhibited the least asynchronism by propagating topics through semantic graphs over multiple text streams.

To further prove that our time synchronization technique of semantic graphs helped to discover more informative topics, we computed the pairwise KL-divergence between the top ten common topics (with the highest average value of $p(z|t)$) as follows:

$$KL(z_1, z_2) = \sum_{w} p(w|z_1) log \frac{p(w|z_1)}{p(w|z_2)} \tag{22}$$

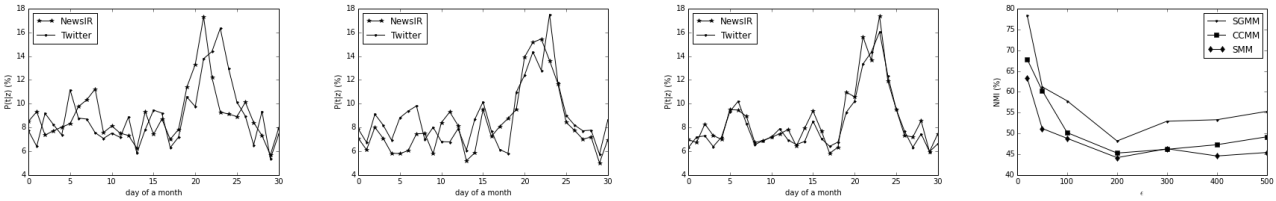
Notice that larger KL-divergence means two topics are more discriminative to each other and 0 divergence means two topics are identical. We present the results in Figure 5, where blue cells represent smaller KL-divergence values and red cells indicate bigger ones. We set $\lambda = 0.6$ when calculating the KL divergence with semantic graph, and $\lambda = 0$ otherwise, all the other parameters were set as the same to Section 5.1. As expected, it is clear that incorporating local semantic graph can discover more discriminative topics than those extracted without the local semantic graphs.

5.4 Performance on Retrieval

As a further demonstration of the utilities of our model, we experimented with twitter retrieval with a similar setup in [12]: we select the top 20 queries from GoogleInsights of UK in the time period of September 2015, corresponding to our datasets, for testing, and the top archive tweets (i.e., search results) returned for each test query were manually labelled as either relevant or not.

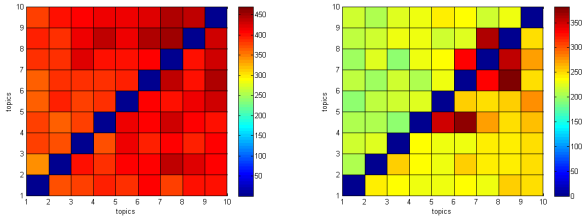
Topic-based Language Model: There could be different topics underlying different queries. In this paper, we propose to take the latent topics into account for twitter retrieval in the language modelling framework:

$$P_{top}(q|d) = \prod_{w \in q} P_{top}(w|d) \tag{23}$$



(a) SMM (b) CCMM (c) SGMM (d) number of K

Figure 4: (a), (b), and (c) show the temporal dynamic of "Rugby World Cup 2015" on NewsIR and Twitter (X-axis is the days in September, 2015. Y-axis is $p(t|z)$) (d) shows the NMI (%) comparison of multiple models with varying K.



(a) SGMM with local semantic graphs (b) SGMM without local semantic graphs

Figure 5: (a) and (b) show the pairwise KL-divergence between topics extracted from multiple streams

$$P_{top}(w|d) = \sum_{k=1}^{N} P(w|z_k)P(z_k|d) \quad (24)$$

$$P_{mix}(q|d) = \alpha P_{cla}(q|d) + \beta P_{top}(q|d) \quad (25)$$

where α and β are two non-negative weight parameters satisfying $\alpha + \beta = 1$, z_k represents latent topics learned from the topic models (see Section4), $P(w|z_k)$ is the unigram language model of topic z_k, and $P(z_k|d)$ is the probability that tweet d belongs to topic z_k.

In our twitter retrieval experiments, we compare the following four approaches:

- the baseline approach which only employs the classic language model (**C**);
- the hybrid approach which combines the classic language model and the *SMM* (**C+S**) [28];
- the hybrid approach which blends the classic language model and the *CCMM* (**C+C**) [12];
- a proposed semantic-based approach which combines the classic language model and the *SGMM* (**C+SG**).

Parameter Setting: All parameter values of these approaches to twitter retrieval were tuned according to Precision at 10 (P@10) [19] or Mean Average Precision (MAP) [19]. In the mixture models (C+S), (C+C) and (C+SG), the ratio between parameter values α and β was set as same as those in [12]. All the other parameters were set to their optimal values that have been found in Section 5.2.

The retrieval performances of those approaches on the test set, measured by $P@10$ and MAP, are reported in Table 3. Consistent to the observation in [12], integrating topics into language model brings substantial performance improvement to the classic language model (C). Moreover, distinguishing common topics from the local ones (C+C) supersedes the model generated from simple mixture model (C+S). More importantly, it is clear that our proposed approach incorporating the semantics-based language model (C+SG) outperforms the other approaches significantly, according to both P@10 and MAP.

Table 3: The experimental results on retrival performance(statistical significance using t-test: ** indicates p-value < 0.01 while * indicates p-value < 0.05).

	C	**C+S**	**C+C**	**C+SG**
P@10	0.275	0.314	0.336	**0.339***
MAP	0.283	0.317	0.343	**0.387****

6. CONCLUSION AND FUTURE WORK

The novel contribution of this paper is in exploiting semantic graphs for the task of topic mining. The performance of our proposed *SGMM* (Semantic Graph based Mixture Model) supersedes the existing ones since it takes account both global semantic graph to overcome the text disjointing problem ((i.e.,lexical gap among the streams)) and local semantic graph to resolve the sources asynchronism problem (i.e.,topic mismatch of different timestamps among the streams). In addition, we have also shown the significant benefit of applying our approach in a twitter retrieval task.

There are several interesting and promising directions in which this work could be extended. First, in this work we only focused on two types of heterogenous sources, namely, Twitter and News, it will be interesting to learn the performance of *SGMM* with multiple streams of varying types. Second,it would be also interesting to investigate the performance of our algorithm by varying the weights of different types of entities. Finally, the parameters are estimated using a simple form of EM algorithm, we would also like to investigate more advanced optimization techniques.

7. ACKNOWLEDGEMENTS

We thank the anonymous reviewer for their helpful comments. We acknowledge support from the EPSRC funded project named **A Situation Aware Information Infrastructure Project** (EP/L026015) . This work was also partly supported by NSF grants #61572223 and #61300144. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the sponsor.

8. REFERENCES

[1] Yang Bao, Nigel Collier, and Anindya Datta. A partially supervised cross-collection topic model for cross-domain text classification. CIKM '13, pages 239–248.

[2] Christian Bizer, Jens Lehmann, Kobilarov, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia - a crystallization point for the web of data. volume 7, pages 154–165.

[3] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty. Latent dirichlet allocation. volume 3, pages 459–565.

[4] Deng Cai, Qiaozhu Mei, Jiawei Han, and Chengxiang Zhai. Modeling hidden topics on document manifold. CIKM '08, pages 911–920.

[5] Pak K. Chan, Martine D. F. Schlag, and Jason Y. Zien. Spectral k-way ratio-cut partitioning and clustering. DAC '93, pages 749–754.

[6] Long Chen, Joemon M. Jose, Haitao Yu, Fajie Yuan, and Dell Zhang. A semantic graph based topic model for question retrieval in community question answering. WSDM '16, pages 287–296.

[7] Xu Chen, Mingyuan Zhou, and Lawrence Carin. The contextual focused topic model. KDD '12, pages 96–104.

[8] Hongbo Deng, Jiawei Han, Bo Zhao, Yintao Yu, and Cindy Xide Lin. Probabilistic topic models with biased propagation on heterogeneous information networks. KDD '11, pages 1271–1279.

[9] Rumi Ghosh and Sitaram Asur. Mining information from heterogeneous sources: A topic modeling approach. MDS-SIGKDD 2013.

[10] Weiwei Guo and Mona Diab. Semantic topic models: Combining word distributional statistics and dictionary definitions. EMNLP '11, pages 552–561.

[11] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. In *Machine Learning*, volume 45, pages 256–269.

[12] Liangjie Hong, Byron Dom, Siva Gurumurthy, and Kostas Tsioutsiouliklis. A time-dependent topic model for multiple text streams. KDD '11, pages 832–840.

[13] Eva Hörster, Rainer Lienhart, and Malcolm Slaney. Image retrieval on large-scale image databases. CIVR '07, pages 17–24.

[14] Ioana Hulpus, Conor Hayes, Marcel Karnstedt, and Derek Greene. Unsupervised graph-based topic labelling using dbpedia. WSDM '13, pages 465–474.

[15] Hyungsul Kim, Yizhou Sun, Julia Hockenmaier, and Jiawei Han. Etm: Entity topic models for mining documents associated with entities. ICDM '12, pages 349–358.

[16] Fang Li, Tingting He, Xinhui Tu, and Xiaohua Hu. Incorporating word correlation into tag-topic model for semantic knowledge acquisition. CIKM '12, pages 1622–1626.

[17] Huajing Li, Zhisheng Li, Wang-Chien Lee, and Dik Lun Lee. A probabilistic topic-based ranking framework for location-sensitive domain information retrieval. SIGIR '09, pages 331–338.

[18] Linlin Li, Benjamin Roth, and Caroline Sporleder. Topic models for word sense disambiguation and token-based idiom detection. ACL '10, pages 1138–1147.

[19] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[20] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[21] Qiaozhu Mei, Deng Cai, Duo Zhang, and Chengxiang Zhai. Topic modeling with network regularization. WWW '08, pages 342–351.

[22] Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. WWW '08, pages 101–110.

[23] Michael Schuhmacher and Simone Paolo Ponzetto. Knowledge-based graph document modeling. WSDM '14, pages 543–552.

[24] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. KDD '04, pages 306–315.

[25] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: Extraction and mining of academic social networks. KDD '08, pages 428–437.

[26] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. WWW '08, pages 111–120.

[27] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: A rating regression approach. KDD '10, pages 783–792.

[28] Xiang Wang, Kai Zhang, Xiaoming Jin, and Dou Shen. Mining common topics from multiple asynchronous text streams. WSDM '09, pages 192–201.

[29] Xiang Wang, Kai Zhang, Xiaoming Jin, and Dou Shen. Mining common topics from multiple asynchronous text streams. WSDM '09, pages 192–201.

[30] Xuanhui Wang, ChengXiang Zhai, Xiao Hu, and Richard Sproat. Mining correlated bursty topic patterns from coordinated text streams. KDD '07, pages 784–793.

[31] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. SIGIR '06, pages 326–335.

[32] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. SIGIR '03, pages 267–273.

[33] ChengXiang Zhai. *Statistical Language Models for Information Retrieval*. Morgan & Claypool Publishers, 2008.

[34] ChengXiang Zhai, Atulya Velivelli, and Bei Yu. A cross-collection mixture model for comparative text mining. KDD '04, pages 743–748.

[35] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. ICML '03, pages 912–919.