Online Sentiment and Topic Dynamics Tracking over the Streaming Data

Yulan He, Chenghua Lin and Amparo Elizabeth Cano
Knowledge Media Institute, The Open University, UK
Email: {y.he, c.lin,amparo.cano}@open.ac.uk

***Abstract*—We propose a dynamic joint sentiment-topic model (dJST) which is able to effectively track sentiment and topic dynamics over the streaming data. Both topic and sentiment dynamics are captured by assuming that the current sentiment-topic specific word distributions are generated according to the word distributions at previous epochs. We study three different ways of accounting for such dependency information, (1) *Sliding window* where the current sentiment-topic-word distributions are dependent on the previous sentiment-topic specific word distributions in the last S epochs; (2) *Skip model* where history sentiment-topic-word distributions are considered by skipping some epochs in between; and (3) *Multiscale model* where previous long- and short- timescale distributions are taken into consideration. We derive efficient online inference procedures to sequentially update the model with newly arrived data and show the effectiveness of our proposed model on the Mozilla add-on reviews crawled between 2007 and 2011.**

I. Introduction

In recent years, social media websites such as Twitter, Facebook, wikis, forums, blogs etc. provide web users venues for expressing and sharing their thoughts and opinions on all kinds of topics. Sentiment analysis on social media allows business organisations to understand their markets and monitor their reputations and brands by extracting online public opinions and sentiments about their products and services. Sentiment dynamics from online contents has been shown to have a strong correlation with fluctuations in macroscopic social and economic indicators in the same time period [1]. Sentiment time series extracted from Twitter messages has also been shown to strongly correlate with polling data on consumer confidence and political opinion [2]. Nevertheless, most existing work detects sentiment in isolation of topic detection and simply records sentiments in different time granularities to form sentiment time series.

Social media data are produced continuously by a large and uncontrolled number of users. The dynamic nature of such data requires the sentiment and topic analysis model to be also dynamically updated, capturing the most recent language use of sentiments and topics in text. In this paper, we propose a dynamic joint sentiment-topic model (dJST) which allows the detection and tracking of views of current and recurrent interests and shifts in topic and sentiment. The dJST model extends from the previously proposed joint sentiment-topic (JST) model which is able to extract coherent and informative topics grouped under different sentiment [3], [4]. The only supervision required by JST learning is domain-independent polarity word prior information.

The previously proposed JST model replies on batch mode learning which assumes that the training data are all available prior to model learning. A limitation of such a batch mode learning approach is that when new data exhibits characteristics significantly different from the existing training data, the model has to be retrained. This is because statistical model performance is proportional to the amount of relevant data samples. On the contrary, incremental or online model learning deals with model structure and/or parameter updating on the arrival of new data. Compared to batch mode model learning, incremental learning tends to be computational efficient as the model is incrementally updated based on the information inferred from the newly arrived data without the need to re-train the model from scratch.

The proposed dJST model, an online counterpart of JST, permits discovering and tracking the intimate interplay between sentiment and topic over time from data. To efficiently handle streaming data, we derive online inference procedures based on a stochastic expectation maximization (EM) algorithm, from which the dJST model can be updated sequentially using the newly arrived data and the parameters of the previously estimated model. Furthermore, to minimize the information loss during the online inference, we assume that the generation of documents in the current epoch is influenced by historical dependencies from the past documents. This is achieved by assuming that the current sentiment-topic specific word distributions are generated from the Dirichlet distribution parameterized by the word-distributions at previous epochs.

While the historical dependencies of past documents can be modeled in many possible ways, we have explored three different time slice settings, namely, the *sliding window* [5], the *skip model* and the *multiscale model*. As the influential power of the historical dependencies may vary over time, we have also investigated two strategies for setting the weights for the historical context at different time slices. These are, to use the exponential decay function and to estimate weights from data directly by EM using the fixed-point iteration method.

The major contribution of this work is three-fold.

- We proposed a dJST model where the generation of documents at current epoch are influenced by documents at historical epochs in three possible ways, (1) *Sliding window* where the current sentiment-topic-word distributions are dependent on the previous sentiment-topic specific word distributions in the last S epochs; (2) *Skip* *model* where history sentiment-topic-word distributions are considered by skipping some epochs in between; and (3) *Multiscale model* where previous long- and short-timescale distributions are taken into consideration.

- We proposed two different weighting strategies to combine documents at historical epochs. One is using an exponential decay function that more recent documents would have a relatively stronger influence on the model parameters in the current epoch compared to earlier documents. Another is to estimate weights from data directly by EM using the fixed-point iteration method [6]. Our experimental results on the Mozilla add-on reviews show that using EM for weights estimation attains better performance than using the exponential decay function.
- We compared the performance of dJST with the two non-dynamic versions of JST, JST-one which only uses the data in the last epoch for training, and JST-all which uses all past data for model learning. The experimental results on the Mozilla add-on reviews show that dJST models outperform JST-one in both perplexity and sentiment classification accuracy which indicates the effectiveness of modelling dynamics. On the other hand, dJST models have much lower perplexities than JST-all. Although they achieve similar sentiment classification accuracies as JST-all, they avoid taking all the historical context into account and hence are computationally more efficient.

We proceed with a review of related work on sentiment and topic dynamics tracking. We then propose the dynamic JST model and describe its online inference procedures as well as the estimation of evolutionary parameters. We demonstrate the effectiveness of our proposed approach by analyzing both sentiment and topic dynamics from review documents crawled from Mozilla review site. Finally, we conclude our work and outline future directions.

II. Related Work

There has been few work on the automatic detection of sentiment dynamics. Mao and Lebanon [7] formulated the sentiment flow detection problem as the prediction of an ordinal sequence based on a sequence of word sets using a variant of conditional random fields based on isotonic regression. Their proposed method has mainly been tested for sentence-level sentiment flow prediction within a document. Mei et al. [8] employed a similar method as in [9] where a hidden Markov model (HMM) is used to tag every word in the collection with a topic and sentiment polarity. The topic life cycles and sentiment dynamics can then be computed by counting the number of words labeled with the corresponding state over time. Their method requires topic and sentiment of each word to be detected beforehand by a topic-sentiment mixture model.

In a recent study, Bollen et al. [1], [10] showed that public mood patterns from a sentiment analysis of Twitter posts do relate to fluctuations in macroscopic social and economic indicators in the same time period. However, they mapped each tweet to a six-dimensional mood vector (Tension, Depression, Anger, Vigour, Fatigue, and Confusion) as defined in the Profile of Mood States (POMS) [11] by simply matching the terms extracted from each tweet to the set of POMS mood adjectives without considering the individual topic each tweet is about.

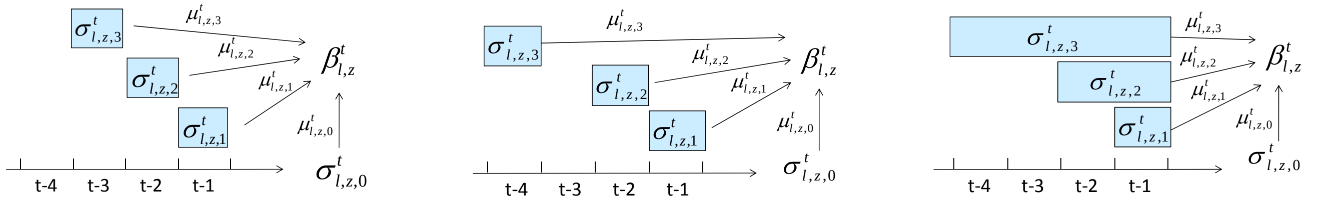
O'Connor et al. [2] extracted tweets messages in relevant to some specific topics and then derived day-to-day sentiment scores by counting positive and negative messages which contain positive or negative words matched against the MPQA subjectivity lexicon[1]. Sentiment time series was generated by smoothing the daily positive vs. negative ratio with a moving average over a window of the past k days. They showed that the smoothed sentiment time series strongly correlated with polling data on consumer confidence and political opinion.

In recent years, there has been a surge of interest in developing topic models to explore topic evolutions over time. The dynamic topic model (DTM) [12] uses a state space model, in particular, the Kalman filter, to capture alignment among topics across different time steps. The continuous time dynamic topic model (cDTM) [13] replaces the discrete state space model of the DTM with its continuous generalization, Brownian motion. While these models employ a Markov assumption over time that the distributions at current epoch only depend on the previous epoch distributions, the topic over time (TOT) model [14] does not make such an assumption, instead, it treats time as an observed continuous variable and for each document, the mixture distribution over topics is influenced by both word co-occurrences and the document's time stamp.

None of the aforementioned models take into account multiscale dynamics. Nallapati et al. [15] proposed the multiscale topic tomography model (MTTM) employs non-homogeneous Poisson processes to model generation of word-counts and models the evolution of topics at various time-scales of resolutions using Haar wavelets. More recently, Iwata et al. [16] proposed online multiscale dynamic topic models (OMDT) which also models the topic evolution with multiple timescales but within the Dirichlet-multinomial framework by assuming current topic-specific distributions over words are generated based on the multiscale word distributions of the previous epoch.

Our work was partly inspired by the previously proposed multiscale topic models [15], [16]. Nevertheless, we have successfully adapted the idea of multiscale modelling for the use in the JST model. We have also additionally proposed another two variants of the dJST model, *sliding window* and *skip model*. Moreover, we have investigated two different ways of setting the weights of evolutionary matrices by either using an exponential decay function or direct estimation from data. As will be discussed in Section IV, setting the weights using the latter method gives superior performance. In addition, both *skip model* and *multiscale model* achieve higher sentiment classification accuracies than *sliding window* although they have similar perplexity results.

[1] http://www.cs.pitt.edu/mpqa/



(a) Sliding window. (b) Skip model. (c) Multiscale model.

Fig. 1. The three dJST variants for $S = 3$. The evolutionary matrix $\boldsymbol{E}^t_{l,z} = [\boldsymbol{\sigma}^t_{l,z,0}, \boldsymbol{\sigma}^t_{l,z,1}, \boldsymbol{\sigma}^t_{l,z,2}, \boldsymbol{\sigma}^t_{l,z,3}]$. The weight matrix $\boldsymbol{\mu}^t_{l,z} = [\mu^t_{l,z,0}, \mu^t_{l,z,1}, \mu^t_{l,z,2}, \mu^t_{l,z,3}]^T$.

III. Dynamic JST (dJST) Model

In a time-stamped document collection, we assume documents are sorted in the ascending order of their time stamps. At each epoch t where the time period for an epoch can be set arbitrarily at, e.g. an hour, a day, or a year, a stream of documents $\{d^t_1, \cdots, d^t_M\}$ of variable size M are received with their order of publication time stamps preserved. A document d at epoch t is represented as a vector of word tokens, $\boldsymbol{w}^t_d = (w^t_{d_1}, w^t_{d_2}, \cdots, w^t_{d_{N_d}})$ where the bold-font variables denote the vectors.

We assume that documents at current epoch are influenced by documents at past. Thus, the current sentiment-topic specific word distributions $\boldsymbol{\varphi}^t_{l,z}$ at epoch t are generated according to the word distributions at previous epochs. In particular, we define an evolutionary matrix of topic z and sentiment label l, $\boldsymbol{E}^t_{l,z}$ where each column is the word distribution of topic z and sentiment label l, $\boldsymbol{\sigma}^t_{l,z,s}$, generated for document streams received within the time slice specified by s which can be set in many different ways. Some of the possible settings are listed below:

- *Sliding window.* If $s \in \{t-S, t-S+1, \cdots, t-1\}$, then this is equivalent to the Markovian assumption that the current sentiment-topic-word distributions are dependent on the previous sentiment-topic specific word distributions in the last S epochs.
- *Skip model.* If $s \in \{t-2^{S-1}, t-2^{S-2} \cdots, t-1\}$, then we are taking history sentiment-topic-word distributions into account by skipping some epochs in between. For example, if $S = 3$, we only consider previous sentiment-topic-word distributions at epoch $t-4$, $t-2$, and $t-1$.
- *Multiscale model.* We could also account for the influence of the past at different timescales to the current epoch [15], [16]. For example, we could set time slice s equivalent to 2^{s-1} epochs. Hence, if $S = 3$, we would consider three previous sentiment-topic-word distributions where the first distribution is between epoch $t-4$ and $t-1$, the second distribution is between epoch $t-2$ and $t-1$, and the third one is at epoch $t-1$. This would allow taking into consideration of previous long- and short- timescale distributions. This model however would take more time and memory spaces and hence effective approximation needs to be performed in order to reduce time/memory complexity.

We then attach a vector of S weights $\boldsymbol{\mu}^t_{l,z} = [\mu^t_{l,z,0}, \mu^t_{l,z,1}, \cdots, \mu^t_{l,z,S}]^T$, each of which determines the contribution of time slice s in computing the priors of $\boldsymbol{\varphi}^t_{l,z}$. Hence, the Dirichlet prior for sentiment-topic-word distributions at epoch t is

$$\boldsymbol{\beta}^t_{l,z} = \boldsymbol{\mu}^t_{l,z} \boldsymbol{E}^t_{l,z} \tag{1}$$

Figure 1 illustrates the three dJST variants proposed here when the number of historical time slices accounted for is set to 3. Here, $\boldsymbol{\sigma}^t_{l,z,s}, s \in \{1..3\}$ is the historical word distribution of topic z and sentiment label l within the time slice specified by s. We set $\boldsymbol{\sigma}^t_{l,z,0}$ for the current epoch as uniform distribution where each element takes the value of $1/(\text{vocabulary size})$.

Assuming we have already calculated the evolutionary parameters $\{\boldsymbol{E}^t_{l,z}, \boldsymbol{\mu}^t_{l,z}\}$ for the current epoch t, the generative dJST model as shown in Figure 2 at epoch t is given as follows:

- For each sentiment label $l = 1, \cdots, L$
  - For each topic $z = 1, \cdots, T$
    - ∗ Compute $\boldsymbol{\beta}^t_{l,z} = \boldsymbol{\mu}^t_{l,z} \boldsymbol{E}^t_{l,z}$
    - ∗ Draw $\boldsymbol{\varphi}^t_{l,z} \sim \text{Dir}(\boldsymbol{\beta}^t_{l,z})$.
- For each document $d = 1, \cdots, D^t$
  - Choose a distribution $\boldsymbol{\pi}^t_d \sim \text{Dir}(\gamma)$.
  - For each sentiment label l under document d, choose a distribution $\boldsymbol{\theta}^t_{d,l} \sim \text{Dir}(\boldsymbol{\alpha}^t)$.
  - For each word $n = 1, \cdots, N_d$ in document d
    - ∗ Choose a sentiment label $l_n \sim \text{Mult}(\boldsymbol{\pi}^t_d)$,
    - ∗ Choose a topic $z_n \sim \text{Mult}(\boldsymbol{\theta}^t_{d,l_n})$,
    - ∗ Choose a word $w_n \sim \text{Mult}(\boldsymbol{\varphi}^t_{l_n,z_n})$.

At epoch 1, the Dirichlet priors $\boldsymbol{\beta}$ of size $L \times T \times V$ are first initialized as symmetric priors of 0.01 [17], and then modified by a transformation matrix $\boldsymbol{\lambda}$ of size $L \times V$ which encodes the word prior sentiment information. $\boldsymbol{\lambda}$ is first initialized with all the elements taking a value of 1. Then for each term $w \in \{1, ..., V\}$ in the corpus vocabulary, the element λ_{lw} is updated as follows

$$\lambda_{lw} = \begin{cases} 0.9 & \text{if } f(w) = l \\ 0.05 & \text{otherwise} \end{cases}, \tag{2}$$

where the function $f(w)$ returns the prior sentiment label of w in a sentiment lexicon, i.e., neutral, positive or negative. For

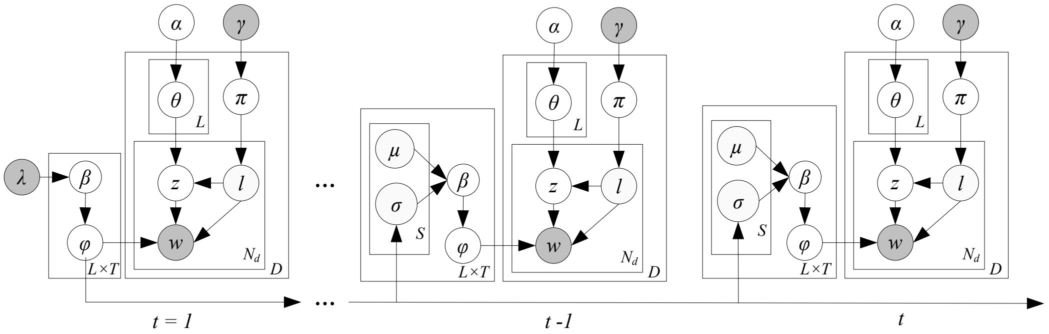


Fig. 2. Dynamic JST model.

example, the word "*excellent*" with index n in the vocabulary has a positive sentiment polarity. The corresponding row vector in $\boldsymbol{\lambda}$ is $[0.05, 0.9, 0.05]$ with its elements representing neutral, positive, and negative prior polarity. For each topic $z \in \{1, ..., T\}$, multiplying λ_{lw} with β_{lzw}, the value of $\beta_{l_{pos}zw}$ is much larger than $\beta_{l_{neu}zw}$ and $\beta_{l_{neg}zw}$. Thus, "*excellent*" has much higher possibility to be drawn from the positive topic word distributions generated from a Dirichlet distribution with parameter $\beta_{l_{pos}}$.

For subsequent epochs, if there are any new words encountered, the word prior polarity information will be incorporated in a similar way. But for existing words, their Dirichlet priors for sentiment-topic-word distributions are obtained using Equation 1.

A. Online Inference

We present a stochastic EM algorithm to sequentially update model parameters at each epoch using the newly obtained document set and the derived evolutionary parameters. At each EM iteration, we infer latent sentiment labels and topics using the collapsed Gibbs sampling and estimate the hyperparameters using maximum likelihood.

The total probability of the model for the document set $\boldsymbol{W}^t$ at epoch t given the evolutionary parameters $\boldsymbol{E}^t, \boldsymbol{\mu}^t$ and the previous model parameter is

$$P(\boldsymbol{W}^t, \boldsymbol{L}^t, \boldsymbol{Z}^t | \gamma^t, \boldsymbol{\alpha}^t, \boldsymbol{E}^t, \boldsymbol{\mu}^t) = \\ P(\boldsymbol{L}^t|\gamma^t) P(\boldsymbol{Z}^t|\boldsymbol{L}^t, \boldsymbol{\alpha}^t) P(\boldsymbol{W}^t|\boldsymbol{L}^t, \boldsymbol{Z}^t, \boldsymbol{E}^t, \boldsymbol{\mu}^t) \quad (3)$$

Gibbs sampling will sequentially sampling each variable of interest, $\boldsymbol{L}^t$ and $\boldsymbol{Z}^t$ here, from the distribution over that variable given the current values of all other variables and the data. Letting the index $x = (d, n, t)$ and the subscript $\backslash x$ denote a quantity that excludes counts in word position n of document d in epoch t, the conditional posterior for z_x and l_x by marginalising out the random variables φ, θ, and π is

$$P(z_x = j, l_x = k | \boldsymbol{W}^t, \boldsymbol{Z}^t_{\backslash x}, \boldsymbol{L}^t_{\backslash x}, \boldsymbol{E}^t, \boldsymbol{\mu}^t) \propto \\ \frac{N^t_{k,j,w\backslash x} + \sum_s \mu^t_{k,j,s} \sigma^t_{k,j,s,w}}{N^t_{k,j\backslash x} + \sum_s \mu^t_{k,j,s}} \cdot \\ \frac{N^t_{d,k,j\backslash x} + \alpha^t_{k,j}}{N^t_{d,k\backslash x} + \sum_j \alpha^t_{k,j}} \cdot \frac{N^t_{d,k\backslash x} + \gamma^t}{N^t_{d\backslash x} + L\gamma^t}, \quad (4)$$

where $N^t_{k,j,w}$ is the number of times word w appeared in topic j and with sentiment label k at epoch t, $N^t_{k,j} = \sum_w N^t_{k,j,w}$, $N^t_{d,k,j}$ is the number of times a word from document d being associated with topic j and sentiment label k at epoch t, $N^t_{d,k}$ is the number of times sentiment label k being assigned to some word tokens in document d at epoch t, $N^t_d = \sum_l N^t_{d,l}$.

We set the symmetric prior $\gamma = (0.05 \times$ average document length$)/L$, where L is the total number of sentiment labels and the value of 0.05 on average allocates 5% of probability mass for mixing. For the asymmetric prior $\boldsymbol{\alpha}$, we initialise it as $\boldsymbol{\alpha}^t = (0.05 \times$ Average document length$)/(L \times T)$ when first entering into a new epoch, where T is the total number of topics. Then for every 40 Gibbs sampling iterations, $\boldsymbol{\alpha}^t$ is updated by

$$(\alpha^t_{l,z})^{\text{new}} \leftarrow \frac{\alpha^t_{l,z} \sum_d [\Psi(N^t_{d,l,z} + \alpha^t_{l,z}) - \Psi(\alpha^t_{l,z})]}{\sum_d [\Psi(N^t_{d,l} + \sum_{z'} \alpha^t_{l,z'}) - \Psi(\sum_{z'} \alpha^t_{l,z'})]}. \quad (5)$$

where $\Psi(\cdot)$ is the digamma function defined by $\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x}$ and $\Gamma(x)$ is a gamma function.

B. Evolutionary Parameters Estimation

There are two sets of evolutionary parameters to be estimated, the weight parameters $\boldsymbol{\mu}$ and the evolutionary matrix $\boldsymbol{E}$. The weight parameters can be set in a way that more recent documents would have a relatively stronger influence on the model parameters in the current epoch compared to earlier documents. One possible setting is an exponential decay function

$$\boldsymbol{\mu}^t = \exp(-\kappa t) \quad (6)$$

which gives the same weight to all the elements in $\boldsymbol{E}^t$. In our experiments, we set $\kappa = 0.5$.

It is also possible to assign different weight to each element in $\boldsymbol{E}^t$ by estimating $\boldsymbol{\mu}$ using the fixed-point iteration method [6] through maximizing the joint distribution in Equation 3. The update formula is:

$$(\mu^t_{l,z,s})^{\text{new}} \leftarrow \frac{\mu^t_{l,z,s} \sum_w \sigma^t_{l,z,s,w} A}{B}, \tag{7}$$

where $A = \Psi(N^t_{l,z,w} + \sum_{s'} \mu^t_{l,z,s'} \sigma^t_{l,z,s',w}) - \Psi(\sum_{s'} \mu^t_{l,z,s'} \sigma^t_{l,z,s',w})$ and $B = \Psi(N^t_{l,z} + \sum_{s'} \mu^t_{l,z,s'}) - \Psi(\sum_{s'} \mu^t_{l,z,s'})$.

The derivation of the evolutionary matrix requires the estimation of each of its elements, $\sigma_{l,z,s,w}$, the word distribution of word w in topic z and sentiment label l at time slice s. This can be defined as follows:

$$\sigma^t_{l,z,s,w} = \frac{C^t_{l,z,s,w}}{\sum_{w'} C^t_{l,z,s,w'}} \tag{8}$$

where $C^t_{l,z,s,w}$ is the expected number of times word w is assigned to sentiment label l and topic z at time slice s. For both the *Sliding window* and *Skip model*, each time slice s only covers a specific epoch t'. Thus $C^t_{l,z,s,w}$ can be obtained directly from the count $\hat{N}^{t'}_{l,z,w}$, i.e., the expected number of times word w is associated with sentiment label l and topic z at epoch t', which can be calculated by

$$\hat{N}^{t'}_{l,z,w} = N^{t'}_{l,z,w} \varphi^{t'}_{l,z,w}, \tag{9}$$

where $N^{t'}_{l,z,w}$ is the observed count for the number of times word w is associated with sentiment label l and topic z at epoch t', and $\varphi^{t'}_{l,z,w}$ is a point estimate of the probability of word w associating with sentiment label l and topic z at epoch t'. In contrast, for the *Multiscale model*, a time slice s might consist of several epochs. Therefore, $C^t_{l,z,w,s}$ is calculated by accumulating the count $\hat{N}^{t'}_{l,z,w}$ over several epochs. The formula for computing $C^t_{l,z,w,s}$ is as follows:

$$C^t_{l,z,s,w} = \begin{cases} \hat{N}^{t'=t-s}_{l,z,w} & \text{Sliding window} \\ \hat{N}^{t'=t-2^{s-1}}_{l,z,w} & \text{Skip model} \\ \sum_{t'=t-2^{s-1}}^{t-1} \hat{N}^{t'}_{l,z,w} & \text{Multiscale model} \end{cases} \tag{10}$$

For this model, the memory requirement increases exponentially with the number of time slices. Following [16], we approximate the update by reducing the frequency for updating long-timescale frequencies that $C^t_{l,z,s,w}$ will only be updated if $t \bmod 2^{s-1} = 0$. This would make sure the memory requirement is linear against the number of time slices.

The Gibbs sampling procedure is given in Algorithm 1.

IV. Experiments

We crawled review documents between March 2007 and January 2011 from the Mozilla Add-ons web site[2]. These reviews are about six different add-ons, Adblock Plus, Video DownloadHelper, Firefox Sync, Echofon for Twitter, Fast Dial, and Personas Plus. All text were downcased and non-English characters were removed. We further pre-processed the documents by stop words removal based on a stop words list[3] and stemming. The final dataset contains 9,114 documents, 11,652 unique words, and 158,562 word tokens in total.

Algorithm 1 Gibbs sampling procedure for dJST.

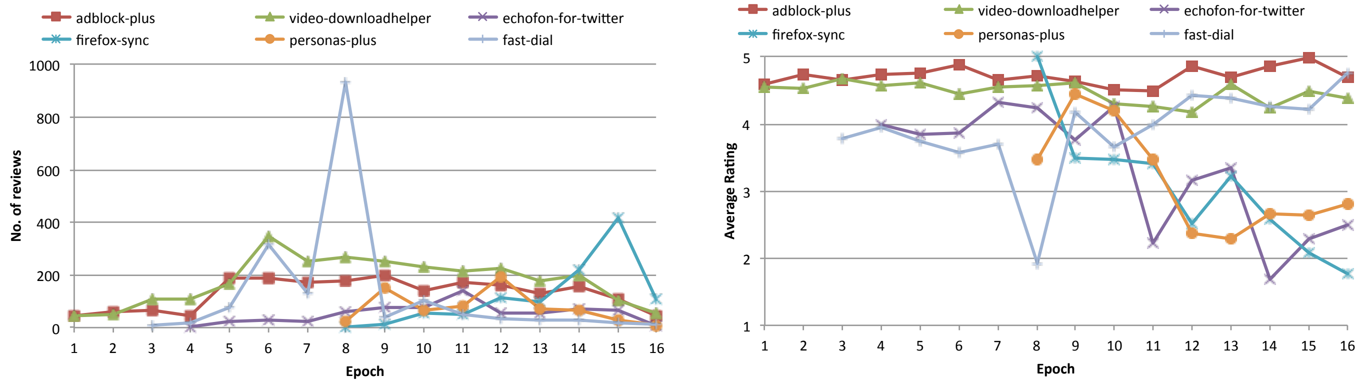
```
Input: Number of topics $T$, number of sentiment labels $L$, number of time slices $S$, Dirichlet prior for document level sentiment distribution $\gamma$, word prior polarity transformation matrix $\boldsymbol{\lambda}$, epoch $t \in \{1, \cdots, \text{maxEpochs}\}$, a stream of documents $D^t = \{d^t_1, \cdots, d^t_M\}$
Output: Dynamic JST model
Sort documents according to their time stamps
for $t = 1$ to maxEpochs do
    if $t == 1$ then
        Set $\boldsymbol{\beta}^t = \boldsymbol{\lambda} \times \mathbf{0.01}$
    else
        Set $\boldsymbol{E}^t_{l,z} = \boldsymbol{E}^{t-1}_{l,z}$
        Set $\boldsymbol{\mu}^t_{l,z} = 1/S$
        Set $\boldsymbol{\beta}^t_{l,z} = \boldsymbol{\mu}^t_{l,z} \boldsymbol{E}^t_{l,z}$
    end if
    Set $\boldsymbol{\alpha}^t = (0.05 \times \text{Average document length})/(L \times T)$
    Initialize $\boldsymbol{\pi}^t, \boldsymbol{\theta}^t, \boldsymbol{\varphi}^t$, and all count variables
    Initialize sentiment label and topic assignment randomly for all word tokens in $D^t$
    for $i = 1$ to $max$ Gibbs Sampling Iterations do
        $[\boldsymbol{\pi}^t, \boldsymbol{\theta}^t, \boldsymbol{\varphi}^t, \boldsymbol{L}^t, \boldsymbol{Z}^t] = \text{GibbsSampling}(D^t, \boldsymbol{\alpha}^t, \boldsymbol{\beta}^t, \boldsymbol{\gamma}^t)$
        for every 40 Gibbs sampling iterations do
            Update $\boldsymbol{\alpha}^t$ using Equation 5
            Update $\boldsymbol{\mu}^t_{l,z}$ using Equation 6 or 7
            Set $\boldsymbol{\beta}^t_{l,z} = \boldsymbol{\mu}^t_{l,z} \boldsymbol{E}^t_{l,z}$
        end for
    end for
    Update $\boldsymbol{E}^t_{l,z}$ using Equation 8
end for
```

The unit epoch was set to quarterly and there were a total of 16 epochs. We plot the total number of reviews for each add-on versus epoch number as shown in Figure 3(a). It can be observed that at the beginning, there were only reviews on Adblock Plus and Video DownloadHelper. Reviews for Fast Dial and Echofon for Twitter started to appear at Epoch 3 and 4 respectively. And reviews on Firefox Sync and Personas Plus only started to appear at Epoch 8. The review occurrence patterns closely relate to the release dates of various add-ons. We also notice that there were a significantly high volume of reviews about Fast Dial at Epoch 8. As for other add-ons, reviews on Adblock Plus and Video DownloadHelper peaked at Epoch 6 while reviews on Firefox Sync peaked at Epoch 15.

Each review is also accompanied with a user rating in the scale of 1 to 5. Figure 3(b) shows the average user rating for each add-on at each epoch. The average user rating across all the epochs for Adblock Plus, Video DownloadHelper, and

[2] https://addons.mozilla.org/

[3] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words/



(a) Number of reviews.

(b) Average user rating.

Fig. 3. Document statistics and average user ratings of reviews for different add-ons.

Firefox Sync are 5-star, 4-star, and 2-star respectively. The reviews of the other three add-ons have an average user rating of 3-star.

We incorporated word polarity prior information into model learning where polarity words were extracted from the two sentiment lexicons, the MPQA subjectivity lexicon and the appraisal lexicon[4]. These two lexicons contain lexical words whose polarity orientations have been fully specified. We extracted the words with strong positive and negative orientation and performed stemming. Duplicate words and words with contradictory polarities after stemming were removed automatically. The final sentiment lexicon consists of 1,511 positive and 2,542 negative words.

A. *Predictive Perplexity*

Perplexity measures a model's prediction ability on unseen data. Lower perplexity implies better predictiveness and hence a better model. We compute the per-word predictive perplexity of the document set D_t at time slice t based on the trained model $\mathcal{M}$

$$\text{perplexity}(t) = \exp\{-\frac{1}{|D_t|}\sum_{d\in D_t}\frac{\log p(\boldsymbol{w}_d|\mathcal{M})}{N_d}\}$$

We evaluate models with perplexity by computing the per-word predictive perplexity of the documents at epoch t based on the data of the previous epochs. We compare the perplexity of dJST with different ways of incorporating historical context into model learning, *sliding window*, *skip model*, and *multiscale model*. For all these models, the weights of the evolutionary matrices are set either based on a decay function (-decay) or estimated directly from data using Equation 7 and denoted as -EM. We set the number of topics to 15 under each of the three sentiment labels, which is equivalent to a total of 45 sentiment-topic clusters. Figure 4 shows the average perplexity over epochs with different number of time slices. It can be observed that increasing the number of time slices results in the decrease of perplexity values, although the decrease in perplexities becomes negligible when the number of time slices is beyond 4. Also, apart from time slice 1, models with their weights of the evolutionary matrices estimated from data using EM give lower perplexities than the models with weights set using the decay function. Hence, in all the subsequent experiments, we estimated the weights of the evolutionary matrices from data using EM unless otherwise specified.

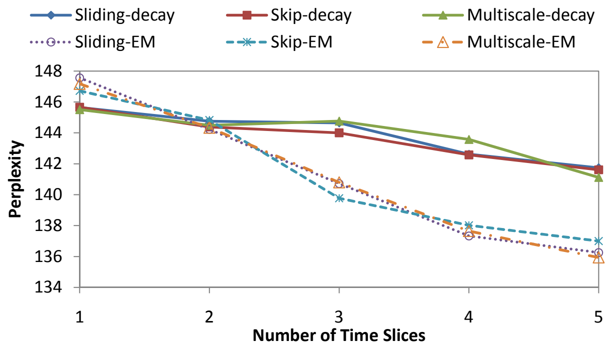


Fig. 4. Perplexity vs number of time slices.

The average perplexity for each epoch with the number of time slices set to 4 and the number of topics set to 15 for the dJST-related models is shown in Figure 5. In addition, we also plot the perplexity results of LDA-one, JST-one, and JST-all. LDA-one and JST-one only use the data in the previous epoch for training and hence it does not model dynamics. JST-all uses all past data for model learning. We set the number of topics to 15 for both JST-one and JST-all. For LDA-one, the number of topics was set to 3 corresponding to positive, negative, and neutral sentiment labels. Word-polarity prior information was incorporated into LDA-one in a similar way as the dJST or JST models[5].

[4] http://lingcog.iit.edu/arc/appraisal_lexicon_2007b.tar.gz

[5] One may argue that the number of topics in LDA should be set to 45, which is equivalent to 15 topics under each of the 3 sentiment labels in JST or dJST models. However, as our task is for both sentiment and topic detection, setting the topic number to 45 makes it difficult to incorporate word polarity prior information into LDA and it is thus not possible to use LDA for document-level sentiment classification.

Figure 5 shows that LDA-one has the highest perplexity values followed by JST-all and JST-one. The perplexity gap between JST-all and the dJST models increases with the increasing number of epochs. This suggests that the dependence of historical reviews vary over time with older reviews having less influence. The variants of dJST models have quite similar perplexities and they all outperform JST-one.

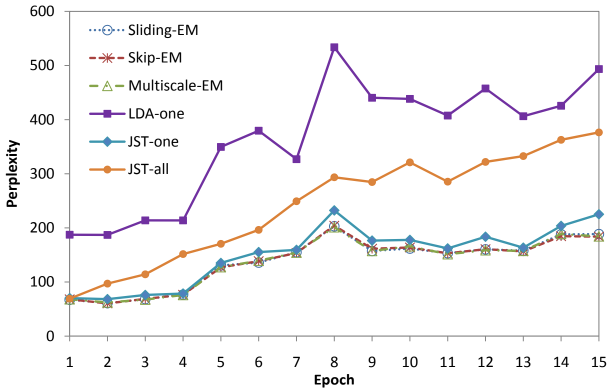


Fig. 5. Perplexity vs number of epochs.

Figure 6 shows the average training time per epoch with the number of topics set to 15 using a computer with a duo core CPU 2.8GHz and 2G memory. Sliding, skip, and multiscale decay models have similar average training time across the number of time slices. For the dJST EM models, estimating the weights of evolutionary matrices takes up more time, with its training time increasing linearly against the number of time slices. JST-one has less training time than the dJST models. LDA-one uses least training time since it only models 3 sentiment topics while others all model a total of 45 sentiment topics. JST-all takes much more time than all the other models as it needs to use all the previous data for training.

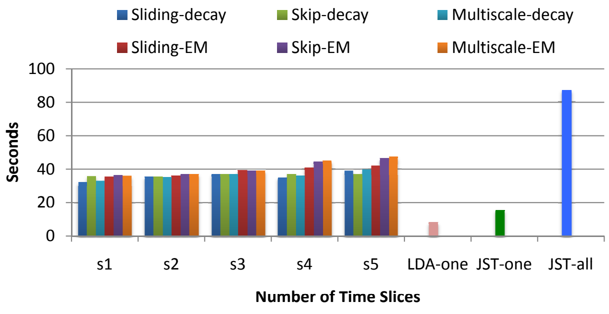
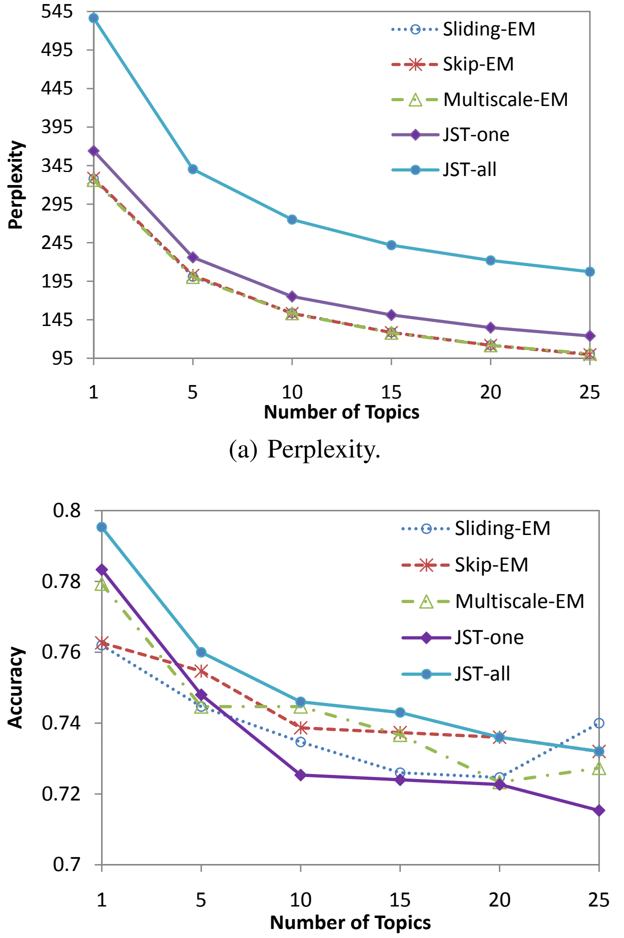


Fig. 6. Average training time per epoch with different number of time slices.

B. Comparison with Other Models

We compare the performance of dJST models with the non-dynamic version of JST models in terms of perplexity and sentiment classification accuracy. For dJST models, we fix the number of time slices to 4. Figure 7(a) shows the average per-word perplexity over epochs with different number of topics. JST-all has higher perplexities than all the other models and the perplexity gap with the dJST models increases with the increased number of topics. All the variants of the dJST model have fairly similar perplexity values and they outperform both JST-all and JST-one.



(a) Perplexity.

(b) Classification accuracy.

Fig. 7. Perplexity and sentiment classification accuracy versus number of topics.

Figure 7(b) shows the average document-level sentiment classification accuracy over epochs with different number of topics. The document-level sentiment classification is based on the probability of sentiment label given a document $P(l|d)$. For the data used here, since each review document is accompanied with a user rating, documents rated as 4 or 5 stars are considered as true positive and other ratings as true negative. This is in contrast to most existing sentiment classification work where reviews rated as 3 stars are removed since they are likely to confuse classifiers. Also, as opposed to most existing approaches, we did not purposely make our dataset balanced (i.e., with the same number of positive and negative documents) for training. dJSTs outperform JST-one with skip-EM and multiscale-EM having similar sentiment classification accuracies as JST-all beyond topic number 1. Also, setting the number of topics to 1 achieves the best classification accuracy for all the models. Increasing the number of topics leads to a

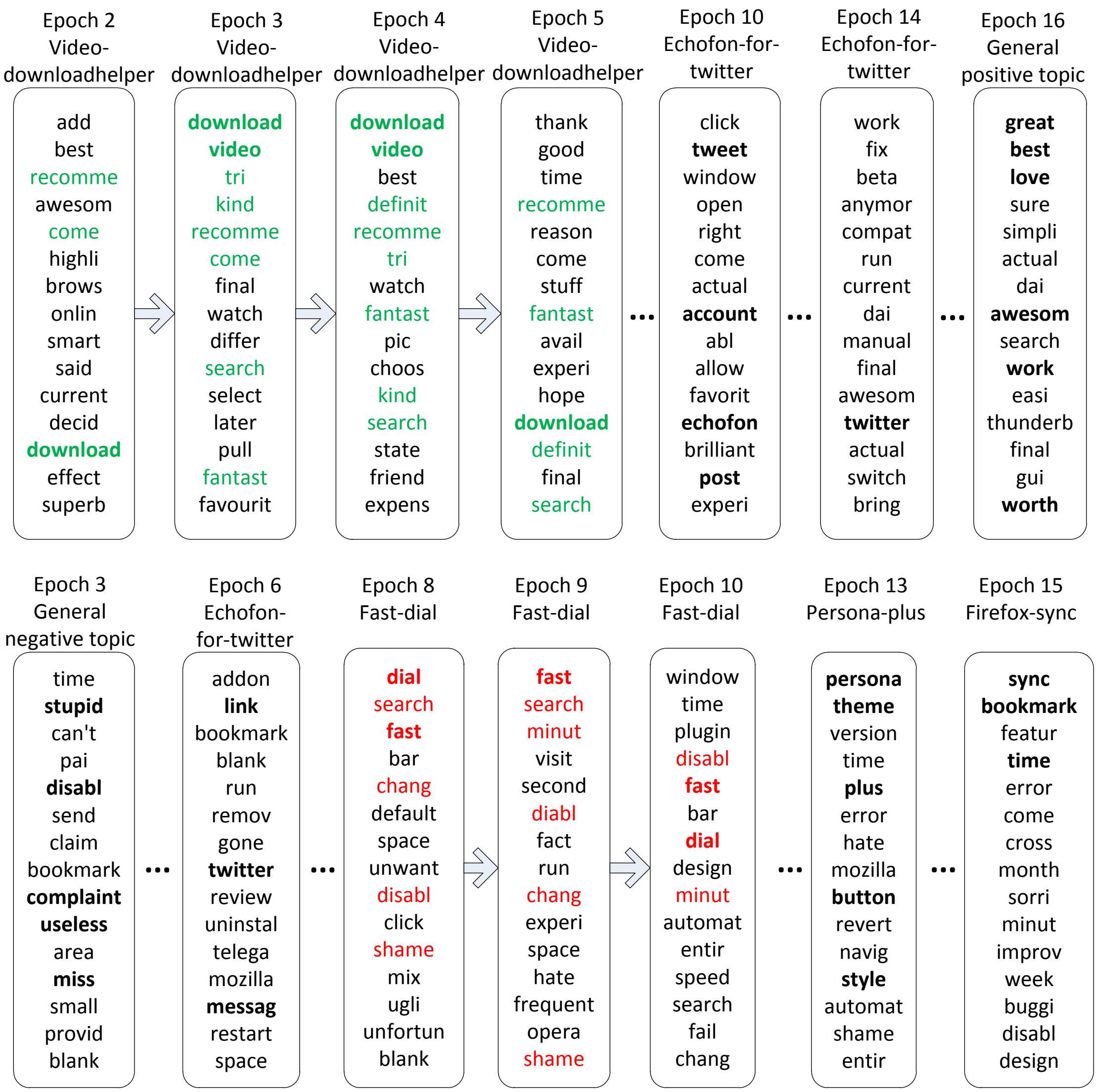


Fig. 8. Example topics evolved over time. Topic labels were derived from bold-face words. The upper and lower panels show the topics under positive and negative sentiment respectively. Words that remain the same in consecutive epochs are highlighted in green or red colors.

slight drop in accuracy though it stabilises at the topic number 10 and beyond for all the models.

In conclusion, both *skip model* and *multiscale model* achieve similar sentiment classification accuracies as JST-all, but they avoid taking all the historical context into account and hence are computationally more efficient. On the other hand, dJST models outperform JST-one in terms of both perplexity values and sentiment classification accuracies which indicates the effectiveness of modelling dynamics.

C. *Example Topics*

Figure 8 shows the evolution of one positive sentiment topic and one negative sentiment topic extracted by dJST-multiscale with the number of topics set to 10 and the number of time slices set to 4. In Figure 9, we plotted the the occurrence probability of these two topics with time, where the probability of a topic z occurred under a sentiment label l, over the document set D_t in each epoch t is calculated as $P(z,l) = \frac{1}{|D_t|}\sum_{d\in D_t} P(z|l,d)P(l|d)$.

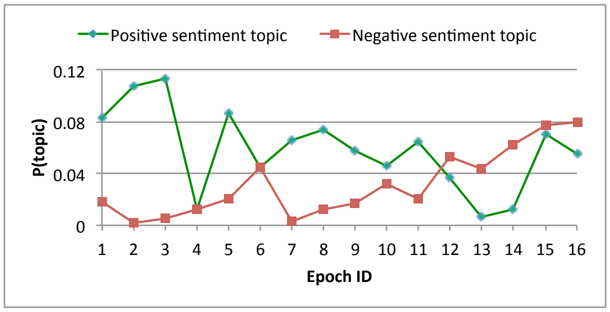


Fig. 9. Occurrence probability of topics with time. Positive and negative sentiment topics correspond to the topics listed in the upper and lower panel of Figure 8 respectively.

We can observe in Figure 8 that the topic-sentiments revealed by the dJST model correlate well with the actual review ratings. At the beginning, the positive sentiment topics were more about Video DownloadHelper (upper panel of Figure 8). Indeed, there are only reviews on Video DownloadHelper or Adblock Plus in the earlier epochs and their average ratings are over 4.5 stars. Figure 9 also shows that the occurrence of this positive topic is more prominent than the negative one in the first 3 epochs. At Epoch 8, there were a significantly high volume of reviews about Fast Dial and the average rating is about 2 stars as shown in Figure 3. We observe that the negative sentiment topics about Fast Dial start to emerge at Epoch 8 (Lower panel of Figure 8). We also see the positive sentiment topic about Echofon for Twitter at Epoch 10, which aligns with the actual average user rating (over 4 stars) on this add-on.

V. Conclusions

In this paper, we have proposed the dynamic joint sentiment-topic (dJST) model which models dynamics of both sentiment and topics over time by assuming that the current sentiment-topic specific word distributions are generated according to the word distributions at previous epochs. We studied three different ways of accounting for such dependency information, sliding window, skip model, and multiscale model, and demonstrated the effectiveness of dJST on a real-world data set in terms of predictive likelihood and sentiment classification accuracy. Our experimental results show that while these three models give similar perplexity values, both the skip model and multiscale model generates better sentiment classification results than sliding window.

Acknowledgment

The authors would like to thank Dong Liu for crawling Mozilla add-ons review data. This work was partially supported by the EPSRC grant EP/J020427/1 and EC-FP7 project ROBUST (grant number 257859).

References

[1] J. Bollen, A. Pepe, and H. Mao, *Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena*, 2010, http://arxiv.org/abs/0911.1583.

[2] B. O'Connor, R. Balasubramanyan, B. Routledge, and N. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010, pp. 122–129.

[3] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *CIKM*, 2009, pp. 375–384.

[4] C. Lin, Y. He, R. Everson, and S. Rueger, "Weakly-supervised Joint Sentiment-Topic Detection from Text," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 1134–1145, 2012.

[5] Y. He, C. Lin, W. Gao, and K. Wong, "Tracking Sentiment and Topic Dynamics from Social Media," in *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2012.

[6] T. Minka, "Estimating a Dirichlet distribution," Tech. Rep., 2003.

[7] Y. Mao and G. Lebanon, "Isotonic conditional random fields and local sentiment flow," in *NIPS*, vol. 19, 2007, pp. 961–968.

[8] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: modeling facets and opinions in weblogs," in *WWW*, 2007, pp. 171–180.

[9] Q. Mei and C. Zhai, "Discovering evolutionary theme patterns from text: an exploration of temporal text mining," in *KDD*, 2005, pp. 198–207.

[10] J. Bollen, H. Mao, and A. Pepe, "Determining the public mood state by analysis of microblogging posts," in *Proceedings of the Alife XII Conference*, 2010.

[11] D. McNair, M. Lorr, and L. Droppleman, *Profile of Mood States: POMS*. EdiTS, Educational and Industrial Testing Service, 1992.

[12] D. Blei and J. Lafferty, "Dynamic topic models," in *ICML*, 2006, pp. 113–120.

[13] C. Wang, D. Blei, and D. Heckerman, "Continuous time dynamic topic models," in *Proc. of UAI*, 2008.

[14] X. Wang and A. McCallum, "Topics over time: a non-Markov continuous-time model of topical trends," in *KDD*, 2006, pp. 424–433.

[15] R. Nallapati, S. Ditmore, J. Lafferty, and K. Ung, "Multiscale topic tomography," in *KDD*, 2007, pp. 520–529.

[16] T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda, "Online multiscale dynamic topic models," in *KDD*, 2010, pp. 663–672.

[17] M. Steyvers and T. Griffiths, "Probabilistic Topic Models," *Handbook of Latent Semantic Analysis*, pp. 427–446, 2007.