Dynamic Joint Sentiment-Topic Model

Yulan He, The Open University
Chenghua Lin, The Open University
Wei Gao, Qatar Foundation
Kam-Fai Wong, The Chinese University of Hong Kong

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. 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. 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.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms: Algorithms,Experimentation

Additional Key Words and Phrases: Dynamic joint sentiment-topic model, Sentiment analysis, Opinion mining, Topic model.

ACM Reference Format:
Y. He et al. 2013. Dynamic Joint Sentiment-Topic Model. ACM Trans. Int. Syst. and Tech. 0, 0, Article 0 (2013), 21 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

The explosive diffusion of the Internet has facilitated the rapid development of a new social phenomena, that of online communities, which exist in almost all areas of society, including social, business, scientific and public service domains. People share their thoughts, express opinions, and seek for support in online communities. 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 [Bollen

This work was partially supported by the EPSRC grant EP/J020427/1, the EC-FP7 project ROBUST (grant number 257859) and the Short Award funded by the Royal Academy of Engineering, UK.
Author's addresses: Y. He and C. Lin, Knowledge Media Institute, The Open University, UK; Email: {y.he,c.lin}@open.ac.uk; Wei Gao, Qatar Computing Research Institute, Qatar Foundation, Qatar; Email: wgao@qf.org.qa; Kam-Fai Wong, Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, China; Email: kfwong@se.cuhk.edu.hk.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

et al. 2010b]. Sentiment time series extracted from Twitter messages has also been shown to strongly correlate with polling data on consumer confidence and political opinion [O'Connor et al. 2010]. Nevertheless, most existing work detects sentiment in isolation of topic detection and simply records sentiments in different time granularity to form sentiment time series.

In this paper, we propose a dynamic joint sentiment-topic model (dJST) [He and Lin 2012] 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 [Lin and He 2009; Lin et al. 2012]. The only supervision required by JST learning is domain-independent polarity word prior information.

The proposal of the dJST model is motivated by two observations. First, the previously proposed JST model assumes that words in text have a static co-occurrence pattern, which may not be suitable for the task of capturing topic and sentiment shifts in a time-variant data corpus. Second, when fitting large-scale data, the standard Gibbs sampling algorithm used in JST can be computationally difficult because it has to repeatedly sample from the posterior the sentiment-topic pair assignment for each word token through the entire corpus at each iteration. The time and memory costs of the batch Gibbs sampling procedure therefore scale linearly with the number of documents analysed.

As an online counterpart of JST, the proposed dJST model addresses the above issues and permits discovering and tracking the intimate interplay between sentiment and topic over time from data. To efficiently fit the model to a large corpus, 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*, 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 four-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 di-

rectly by EM using the fixed-point iteration method [Minka 2003]. 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 show that the dJST models outperform JST-one in both perplexity and sentiment classification accuracy which indicates the effectiveness of modeling dynamics. On the other hand, the 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 explored the impact of different input features on the dJST performance. In particular, we performed part-of-speech (POS) tagging and syntactic parsing and then removed less informative words based on their POS tags and augmented the bag-of-words features with nominal phrases. Our experimental results show that using the new input features improves the sentiment classification accuracy and the topics extracted are generally more meaningful than those from the bag-of-words representations.

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 and the setting of hyperparameters. 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.

2. RELATED WORK

There has been few work on the automatic detection of sentiment dynamics. Mao and Lebanon [2007; 2009] 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. [2007] employed a similar method as in [Mei and Zhai 2005] 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. [2010b; 2010a] 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) [McNair et al. 1992] 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. [2010] 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

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

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) [Blei and Lafferty 2006] 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) [Wang et al. 2008] 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 [Wang and McCallum 2006] 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. [2007] 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. [2010] 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 [Nallapati et al. 2007; Iwata et al. 2010]. 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 5, 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.

Aside from extension of topic models, there have also been increasing interests in incorporating time dependencies into hierarchical Dirichlet process (HDP) [Teh et al. 2006] for revealing topic dynamics from time-stamped documents. One advantage over topic model-based approaches is that HDP allows the automatic discovery of topic numbers. Ren et al. [2008] proposed the dynamic hierarchical Dirichlet process (dHDP) model which imposes a dynamic time dependence so that the initial mixture model and the subsequent time-dependent mixtures share the same set of components. Pruteanu-Malinici et al. [2009] developed a simplified form of dHDP that assumes documents at a given time have topics drawn from a mixture model and the mixture weights over topics evolve with time. Zhang et al. [2010] proposed using a series of HDPs with time dependencies to the adjacent epochs being added to discover cluster evolution patterns from multiple correlated time-varying text corpora. This falls into evolutionary clustering [Chakrabarti et al. 2006; Chi et al. 2007; Ahmed and Xing 2008; Xu et al. 2008b; 2008a; Chi et al. 2009] which aims to generate clusters that fit the data at each epoch as much as possible and at the same time preserves the smoothness of clustering results over time.

3. 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 docu-

Table I. Notations used in the paper.

Symbol	*Description*
D^t	number of documents in epoch t
N_d^t	number of words in document d at epoch t
L	number of sentiment labels
T	number of topics
V	number of unique words in the current epoch
S	number of time slices
γ	symmetric prior for sentiment labels
$\boldsymbol{\alpha}^t$	matrix of $L \times T$ dimension, row l represents the priors of the mixing proportion of topics in sentiment label l
$\boldsymbol{\beta}^t$	matrix of $L \times T \times V$ dimension, priors for the word distribution conditioned on sentiment labels and topics
$\boldsymbol{\pi}_d^t$	parameter notation for the sentiment label mixture proportion for document d^t. $\boldsymbol{\pi}^t = \{\pi_d^t\}_{d=1}^{D^t}$ ($D^t \times L$ matrix)
$\boldsymbol{\theta}_{d,l}^t$	multinomial distribution over topics for the lth sentiment label for document d^t, $\boldsymbol{\theta}^t = \{\{\theta_{d,l}^t\}_{l=1}^L\}_{d=1}^{D^t}$ ($D^t \times L \times T$ matrix)
$\boldsymbol{\varphi}_{l,z}^t$	multinomial distribution over words for the lth sentiment label and zth topic at epoch t. $\boldsymbol{\varphi}^t = \{\{\varphi_{l,z}\}_{z=1}^T\}_{l=1}^L$ ($L \times T \times V$ matrix)
$\boldsymbol{\lambda}$	matrix of $L \times V$ dimension which encodes the word prior sentiment polarity information
$\boldsymbol{E}_{l,z}^t$	Evolutionary matrix of sentiment label l and topic z at epoch t, column size is determined by the total number of time slices taken into account when estimating the prior for the sentiment-topic-word distribution of current epoch
$\boldsymbol{\mu}_{l,z}^t$	weight vector, $\boldsymbol{\mu}_{l,z}^t = [\mu_{l,z,1}^t, \cdots, \mu_{l,z,S}^t]$, each of which determines the contribution of time slice s in computing the priors of $\varphi_{l,z}^t$
$\boldsymbol{\sigma}_{l,z,s}^t$	multinomial word distribution of sentiment label l and topic z with time slice s at epoch t, $\boldsymbol{\sigma}_{l,z,s}^t = \{\sigma_{l,z,s,w}^t\}_{w=1}^V$

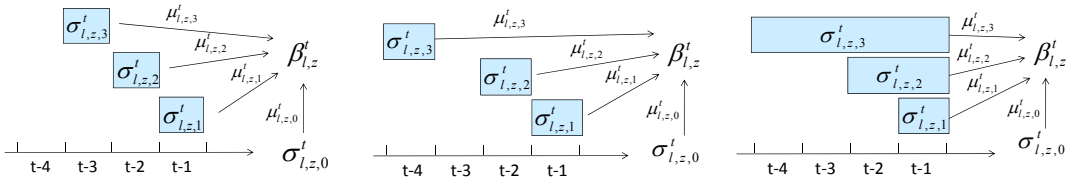
ments $\{d_1^t, \cdots, d_M^t\}$ of variable size M are received with their order of publication time stamps preserved. A document d at epoch t is represented as a vector of word tokens, $\boldsymbol{w}_d^t = (w_{d_1}^t, w_{d_2}^t, \cdots, w_{d_{N_d}}^t)$ where the bold-font variables denote the vectors. Our notations are summarized in Table I.

We assume that documents at current epoch are influenced by documents at past. Thus, the current sentiment-topic specific word distributions $\boldsymbol{\varphi}_{l,z}^t$ 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, $\boldsymbol{E}_{l,z}^t$ where each column is the word distribution of topic z and sentiment label l, $\boldsymbol{\sigma}_{l,z,s}^t$, 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 [Nallapati et al. 2007; Iwata et al. 2010]. 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.

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.



(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 $\boldsymbol{\mu}^t_{l,z} = [\mu^t_{l,z,0}, \mu^t_{l,z,1}, \mu^t_{l,z,2}, \mu^t_{l,z,3}]^T$.

We then attach a vector of $S+1$ weights $\boldsymbol{\mu}^t_{l,z} = \{\mu^t_{l,z,s}\}^S_{s=0}$ ($\mu^t_{l,z,s} > 0, \sum^S_{s=0} \mu^t_{l,z,s} = 1$) with its components representing the weights that each time slice s contributes to calculating the priors of $\boldsymbol{\varphi}^t_{l,z}$. Particularly, we set $\{\sigma^{t-1}_{l,z,0,w}\}^V_{w=1} = 1/V$ for the time scale $s = 0$ as a form of smoothing to avoid the zero probability problem for unseen words, where V is the number of unique words in the documents.

The Dirichlet prior for sentiment-topic-word distributions at epoch t is

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

The current sentiment-topic word distributions $\boldsymbol{\varphi}^t_{l,z}$ at epoch t are generated from the Dirichlet distribution parameterized by $\boldsymbol{\beta}^t_{l,z}$, $\boldsymbol{\varphi}^t_{l,z} \sim \text{Dir}(\boldsymbol{\beta}^t_{l,z})$. With this formulation, we can ensure that the mean of the Dirichlet parameter for the current epoch becomes proportional to the weighted sum of the word distributions at previous epochs.

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 $w_n \sim \text{Mult}(\boldsymbol{\varphi}^t_{l_n,z_n})$.

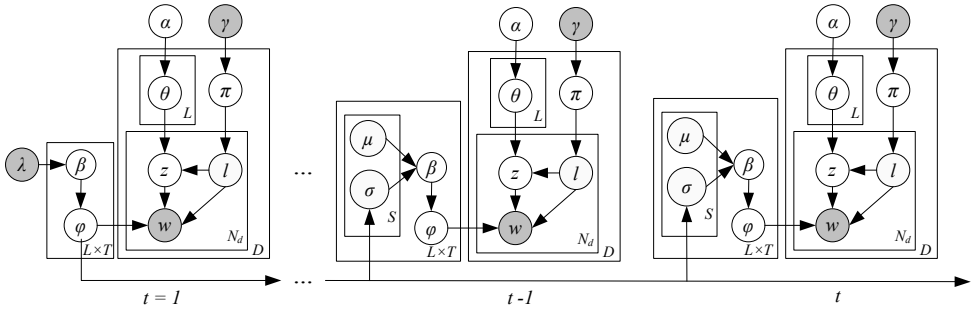


Fig. 2. Dynamic JST model.

3.1. 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

$$P(\boldsymbol{W}^t, \boldsymbol{L}^t, \boldsymbol{Z}^t|\gamma^t, \boldsymbol{\alpha}^t, \boldsymbol{E}^t, \boldsymbol{\mu}^t) = P(\boldsymbol{L}^t|\gamma^t)P(\boldsymbol{Z}^t|\boldsymbol{L}^t, \boldsymbol{\alpha}^t)P(\boldsymbol{W}^t|\boldsymbol{L}^t, \boldsymbol{Z}^t, \boldsymbol{E}^t, \boldsymbol{\mu}^t) \tag{2}$$

For the first term on the RHS of Equation 2, by integrating out π, we obtain

$$P(\boldsymbol{L}^t|\gamma^t) = \prod_d \frac{\Gamma(L\gamma^t)}{\Gamma(\gamma^t)^L} \frac{\prod_l \Gamma(N_{d,l}^t + \gamma^t)}{\Gamma(N_d^t + L\gamma^t)}, \tag{3}$$

where D is the total number of documents in epoch t, $N_{d,l}^t$ is the number of times sentiment label l being assigned to some word tokens in document d at epoch t, $N_d^t = \sum_l N_{d,l}^t$, and Γ is the gamma function.

For the second term, by integrating out θ, we obtain

$$P(\boldsymbol{Z}^t|\boldsymbol{L}^t, \boldsymbol{\alpha}^t) = \prod_d \prod_l \frac{\Gamma(\sum_{z=1}^T \alpha_{l,z}^t)}{\prod_{z=1}^T \Gamma(\alpha_{l,z}^t)} \frac{\prod_z \Gamma(N_{d,l,z}^t + \alpha_{l,z}^t)}{\Gamma(N_{d,l}^t + \sum_z \alpha_{l,z}^t)}, \tag{4}$$

where $N_{d,l,z}^t$ is the number of times a word from document d being associated with topic z and sentiment label l at epoch t, and $N_{d,l}^t = \sum_z N_{d,l,z}^t$.

For the last term, by integrating out φ, we obtain

$$P(\boldsymbol{W}^t|\boldsymbol{L}^t, \boldsymbol{Z}^t) = \prod_l \prod_z \frac{\Gamma(\sum_s \mu_{l,z,s}^t)}{\prod_w \Gamma(\sum_s \mu_{l,z,s}^t \sigma_{l,z,s,w}^t)} \frac{\prod_w \Gamma(N_{l,z,w}^t + \sum_s \mu_{l,z,s}^t \sigma_{l,z,s,w}^t)}{\Gamma(N_{l,z}^t + \sum_s \mu_{l,z,s}^t)}, \tag{5}$$

where $N_{l,z,w}^t$ is the number of times word w appeared in topic z and with sentiment label l at epoch t, $N_{l,z}^t = \sum_w N_{l,z,w}^t$.

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

$$P(z_x = j, l_x = k|\boldsymbol{W}^t, \boldsymbol{Z}^t_{\setminus x}, \boldsymbol{L}^t_{\setminus x}, \boldsymbol{E}^t, \boldsymbol{\mu}^t) \propto \frac{N^t_{k,j,w_j\setminus x} + \sum_s \mu^t_{k,j,s}\sigma^t_{k,j,s,w_j}}{N^t_{k,j\setminus x} + \sum_s \mu^t_{k,j,s}} \cdot \frac{N^t_{d,k,j\setminus x} + \alpha^t_{k,j}}{N^t_{d,k\setminus x} + \sum_j \alpha^t_{k,j}} \cdot \frac{N^t_{d,k\setminus x} + \gamma^t}{N^t_{d\setminus x} + L\gamma^t}. \quad (6)$$

3.2. Evolutionary Parameters Estimation

There are two sets of evolutionary parameters to be estimated, the weight parameters $\boldsymbol{\mu}$ and the evolutionary matrix $\boldsymbol{E}$.

3.2.1. Estimating the Weight Vector $\boldsymbol{\mu}^t$. We have explored two different strategies for setting $\boldsymbol{\mu}^t$. These are using the exponential decay function and learning the weight directly from data using the fixed-point iteration method.

Exponential Decay Function 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

$$\boldsymbol{\mu}^t = \exp(-\kappa t) \quad (7)$$

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

Fixed-point Iteration It is also possible to estimate the weight vector $\boldsymbol{\mu}^t$ directly from data by maximizing the joint distribution in Equation 2 using the fixed-point iteration method [Minka 2003]. The update formula is:

$$(\mu^t_{l,z,s})^{\text{new}} \leftarrow \frac{\mu^t_{l,z,s}\sum_w \sigma^t_{l,z,s,w} A}{B}, \quad (8)$$

where

$$A = \Psi(N^t_{l,z,w} + \sum_{s'} \mu^t_{l,z,s'}\sigma^t_{l,z,s',w}) - \Psi(\sum_{s'} \mu^t_{l,z,s'}\sigma^t_{l,z,s',w}), \quad (9)$$

$$B = \Psi(N^t_{l,z} + \sum_{s'} \mu^t_{l,z,s'}) - \Psi(\sum_{s'} \mu^t_{l,z,s'}), \quad (10)$$

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

The detailed derivation of the update formula for $\boldsymbol{\mu}$ is presented in Appendix A.

3.2.2. Estimating the Evolutionary Matrix $\boldsymbol{E}^t$. The evolutionary matrix $\boldsymbol{E}^t$ accounts for the historical word distributions at different time slices. The derivation of $\boldsymbol{E}^t$ therefore requires the estimation of each of its elements, $\{\sigma^t_{l,z,s,w}\}^V_{w=1}$, the word distribution in topic z and sentiment label l at time slice s, which can be calculated as follows:

$$\sigma^t_{l,z,s,w} = \frac{C^t_{l,z,s,w}}{\sum_w C^t_{l,z,s,w}}, \quad (11)$$

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

$$\hat{N}^{t'}_{l,z,w} = N^{t'}_{l,z,w}\varphi^{t'}_{l,z,w}, \quad (12)$$

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' recovered using Equation 13.

$$\varphi^{t}_{k,j,i} = \frac{N^{t}_{k,j,i} + \sum_s \mu^{t}_{k,j,s}\sigma^{t}_{k,j,s,i}}{N^{t}_{k,j} + \sum_s \mu^{t}_{k,j,s}}. \tag{13}$$

For the *Multi-scale 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:

$$C^{t}_{l,z,s,w} = \begin{cases} \hat{N}^{t'=t-s}_{l,z,w} & \text{Sliding window} \\ \hat{N}^{t'=t-2^{s-1}}_{l,z,w} & \text{Skip model} \\ \sum_{t'=t-2^{s-1}}^{t-1} \hat{N}^{t'}_{l,z,w} & \text{Multi-scale model} \end{cases} \tag{14}$$

where the value of s ranges from 1 to S, the total number of historical time slices to be accounted.

3.3. Hyperparameter Settings

The dJST models consist of three hyperparameters, $\boldsymbol{\alpha}^t$, $\boldsymbol{\beta}^t$ and γ^t. We estimated $\boldsymbol{\alpha}^t$ from data using maximum-likelihood as part of the online stochastic EM algorithm and set both $\boldsymbol{\beta}^t$ and γ^t empirically.

Setting $\boldsymbol{\alpha}^t$ A common practice for the implementations of topic models is to use symmetric Dirichlet hyperparameters. However, it has been found that an asymmetric Dirichlet prior over the per-document topic proportions has substantial advantages over a symmetric prior [Wallach et al. 2009]. So when first entering a new epoch, we initialize the asymmetric $\boldsymbol{\alpha}^t = (0.05 \times \text{avgDocLength}^t/(L \times T)$, where avgDocLength^t is the average document length of epoch t and the value of 0.05 on average allocates 5% of probability mass for mixing. Afterwards for every 40 Gibbs sampling iterations, $\boldsymbol{\alpha}^t$ is learned directly from data using maximum-likelihood estimation [Minka 2003]

$$(\alpha^{t}_{l,z})^{\text{new}} \leftarrow \frac{\alpha^{t}_{l,z} \sum_d [\Psi(N^{t}_{d,l,z} + \alpha^{t}_{l,z}) - \Psi(\alpha^{t}_{l,z})]}{\sum_d [\Psi(N^{t}_{d,l} + \sum_{z'} \alpha^{t}_{l,z'}) - \Psi(\sum_{z'} \alpha^{t}_{l,z'})]}. \tag{15}$$

Setting $\boldsymbol{\beta}^t$ At epoch 1, the Dirichlet prior $\boldsymbol{\beta}$ of size $L \times T \times V$ is first initialized as symmetric priors of 0.01 [Steyvers and Griffiths 2007], 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

$$\lambda_{lw} = \begin{cases} 0.9 & \text{if } f(w) = l \\ 0.05 & \text{otherwise} \end{cases}, \tag{16}$$

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

In our work here, 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[2]. 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.

Setting γ^t We empirically set the symmetric prior $\gamma^t = (0.05 \times \text{avgDocLength}^t)/L$, where the value of 0.05 on average allocates 5% of probability mass for mixing.

The complete procedures for the online stochastic EM algorithm for the dJST model is given in Algorithm 1.

4. EXPERIMENTAL SETUP

4.1. Dataset

We crawled review documents between March 2007 and January 2011 from the Mozilla Add-ons web site[3]. 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 preprocessed the documents by stop words removal based on a stop words list[4] and stemming. The final dataset contains 9,114 documents, 11,652 unique words, and 158,562 word tokens in total.

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. We also notice that there were a significantly high volume of reviews about Fast Dial at Epoch 8. As for the 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 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.

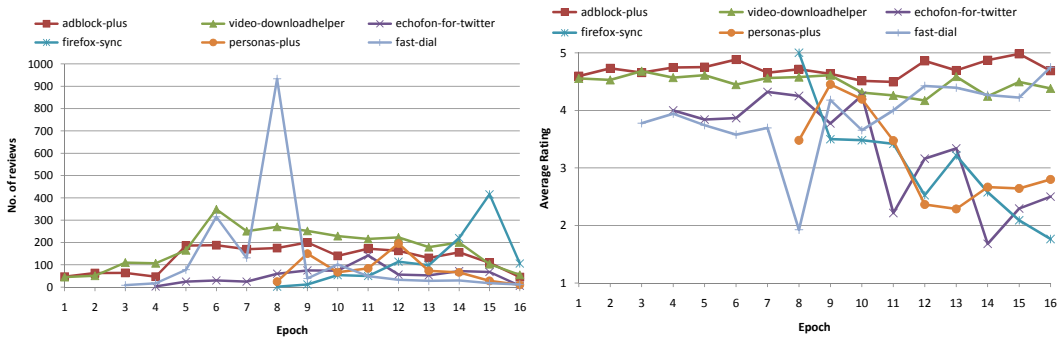
4.2. Evaluation Metrics

We evaluate the dJST model performance in terms of predictive perplexity and document-level sentiment classification accuracy, which are defined as follows.

[2]http://lingcog.iit.edu/arc/appraisal_lexicon_2007b.tar.gz
[3]https://addons.mozilla.org/
[4]http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words/

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 γ, word prior polarity transformation matrix $\boldsymbol{\lambda}$, epoch $t \in \{1, \cdots, \text{maxEpochs}\}$, a stream of documents $D^t = \{d_1^t, \cdots, d_M^t\}$
Output: Dynamic JST model
Sort documents according to their time stamps;
for $t = 1$ *to maxEpochs* **do**
  if t == *1* **then**
    Set $\boldsymbol{\beta}^t = \boldsymbol{\lambda} \times \mathbf{0.01}$;
  end
  else
    Set $\boldsymbol{E}_{l,z}^t = \boldsymbol{E}_{l,z}^{t-1}$;
    Set $\boldsymbol{\mu}_{l,z}^t = 1/S$;
    Set $\beta_{l,z}^t = \boldsymbol{\mu}_{l,z}^t \boldsymbol{E}_{l,z}^t$;
  end
  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 15 ;
      Update $\boldsymbol{\mu}_{l,z}^t$ using Equation 7 or 8 ;
      Set $\beta_{l,z}^t = \boldsymbol{\mu}_{l,z}^t \boldsymbol{E}_{l,z}^t$;
    end
    for *every 200 Gibbs sampling iterations* **do**
      Update $\boldsymbol{\Pi}^t, \boldsymbol{\Theta}^t, \boldsymbol{\Phi}^t$ with the new sampling results ;
    end
  end
  Update $\boldsymbol{E}_{l,z}^t$ using Equation 11 ;
end



(a) Number of reviews. (b) Average user rating.

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

Predictive Perplexity Originally used in language modelling, perplexity measures a model's prediction ability on unseen data. It is defined as the reciprocal geometric mean of the likelihood of a test corpus given a trained model's Markov chain state $\mathcal{M}$. Lower perplexity implies better predictiveness, and hence a better model. In the dJST

experiments, we computed the per-word predictive perplexity of the unseen test set $\tilde{\boldsymbol{\mathcal{D}}}_t = \{\tilde{\boldsymbol{w}}_d^t\}_{d=1}^{D^t}$ at epoch t based on the previously trained model $\boldsymbol{\mathcal{M}} = \{\mathbf{w}, \mathbf{z}, \mathbf{l}\}$ as

$$\text{Perplexity} = P(\tilde{\boldsymbol{\mathcal{D}}}_t|\boldsymbol{\mathcal{M}}) = \exp\{-\frac{\sum_{d=1}^{D^t} \log p(\tilde{\boldsymbol{w}}_d^t|\boldsymbol{\mathcal{M}})}{\sum_{d=1}^{D^t} \tilde{N}_d^t}\}, \tag{17}$$

where

$$P(\tilde{\boldsymbol{w}}_d^t|\boldsymbol{\mathcal{M}}) = \prod_{n=1}^{\tilde{N}_d^t} \prod_{l=1}^{L} \prod_{z=1}^{T} P(\tilde{w}_{d,n}|l,z)P(z|l)P(l), \tag{18}$$

and $\tilde{\boldsymbol{w}}_d^t$ represents the word vector of the dth document in the test set, and $\tilde{N}_d^t$ is the total number of words in $\tilde{\boldsymbol{w}}_d^t$. Directly expressing the likelihood of the test corpus $P(\tilde{\boldsymbol{w}}_d^t|\boldsymbol{\mathcal{M}})$ as a function of the multinomial parameters $\{\boldsymbol{\Pi}, \boldsymbol{\Theta}, \boldsymbol{\Phi}\}$ of model $\boldsymbol{\mathcal{M}}$ yields,

$$P(\tilde{\boldsymbol{w}}_d^t|\boldsymbol{\mathcal{M}}) = \prod_{i=1}^{V} (\sum_{l=1}^{l} \sum_{z=1}^{T} \varphi_{l,z,i} \cdot \theta_{d,l,z} \cdot \pi_{d,l})^{\tilde{N}_{d,i}^t}, \tag{19}$$

where $\tilde{N}_{d,i}^t$ is the number of times term i has appeared in the dth document of the test set. Using Equation 17 and 19, the perplexity of unseen documents can then be calculated given a trained dJST model.

Sentiment Classification 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.

5. EXPERIMENTAL RESULTS

In this section, we present the experimental results of the dJST model on the Mozilla add-on review dataset.

5.1. Number of Time Slices

dJST accounts for the historical context at previous epochs specified by a total number of S time slices. A larger number of time slices indicate a longer history period modeled by dJST. In order to investigate the influence of the historical time slice on the model performance, we vary $S \in \{1..5\}$ and evaluate the model performance in perplexity. In our experiments, a model trained on the data at epoch $t-1$ is tested on the data of the next epoch t.

We compare the performance 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 8 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

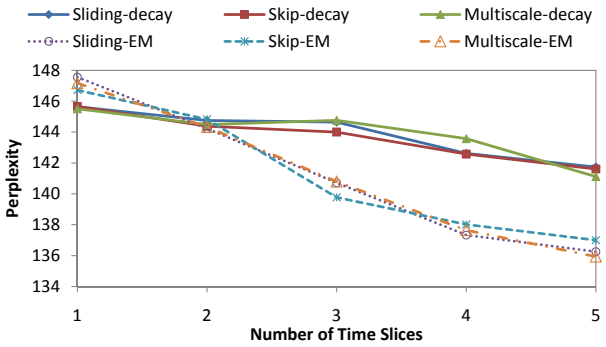


Fig. 4. Perplexity vs number of time slices.

their weights of the evolutionary matrices estimated from data using EM give lower perplexities than the models with weights set using the decay function. In all the subsequent experiments, we estimated the weight vector of the evolutionary matrix from data using EM unless otherwise specified.

5.2. Comparison with Other Models

In order to evaluate the effectiveness of dJST in modelling dynamics, we compare the performance of the dJST models in terms of perplexity and sentiment classification accuracy with the non-dynamic version of LDA and JST, namely, LDA-one, JST-one, and JST-all. LDA-one and JST-one only use the data in the previous epoch for training and hence they do not model dynamics, whereas JST-all uses all the past data for model learning.

5.2.1. Perplexity for each epoch. 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].

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.

5.2.2. Performance vs. Different Number of Topics. In another set of experiments we studied the influence of the topic number settings on the dJST model performance. With the number of time slices fixed at $S = 4$, we vary the topic number $T \in \{1, 5, 10, 15, 20, 25\}$. Figure 6(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

[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.

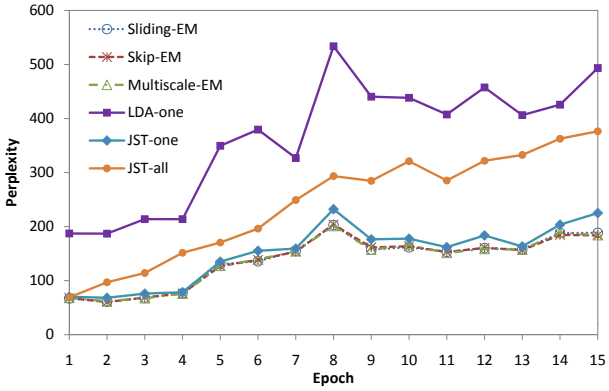
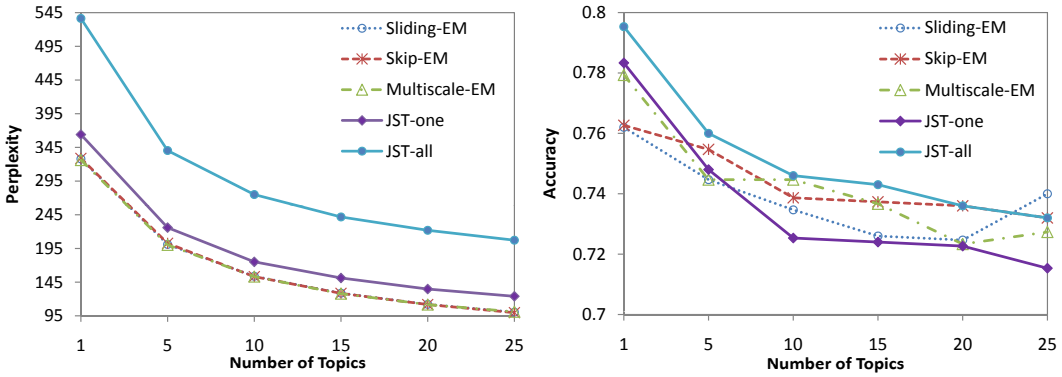


Fig. 5. Perplexity vs number of epochs.



(a) Perplexity. (b) Classification accuracy.

Fig. 6. Perplexity and sentiment classification accuracy versus number of topics.

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.

Figure 6(b) shows the average document-level sentiment classification accuracy over epochs with different number of topics. 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 slight drop in accuracy though it stabilises at the topic number 10 and beyond for all the models. Nevertheless, the drop in sentiment classification accuracy by modelling more topics is only marginal (about 1% drop) for sliding-EM and skip-EM.

5.2.3. Computational Time. Figure 7 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 mod-

els. 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.

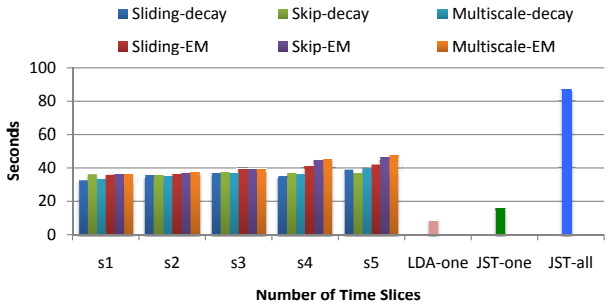
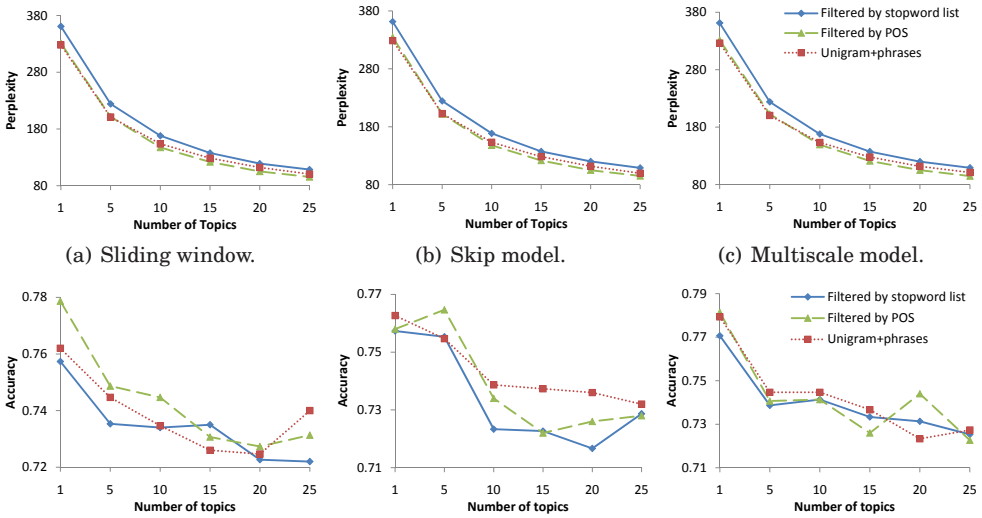


Fig. 7. Average training time per epoch with different number of time slices.

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.

5.3. Exploring Different Input Features



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

(d) Sliding window. (e) Skip model. (f) Multiscale model.

Fig. 8. Performance vs. different input features. Top panel: perplexity; bottom panel: sentiment classification accuracy.

In the previous experiments, we pre-processed documents by removing stop words from a stop word list and used unigrams as input features to model learning. We further conducted another set of experiments by first performing part-of-speech (POS) tagging and syntactic parsing, and then removing words based on their POS tags and

augmenting the bag-of-word features with nominal phrases. We manually constructed a set of 19 POS tags to be ignored, such as PREP (preposition), DET (determiner), PUNC (punctuation), etc. Words with the POS tags falling into such a list were removed. We compare the performance of the dJST models using the original feature representation (*Filtered by stopword list*), by removing words based on POS tags (*Filtered by POS*), and augmenting the bag-of-words feature space with nominal phrases (*Unigrams+phrases*). In the results presented in Figure 8, we set the number of time slices to 4 and topics$\in \{1, 5, 10, 15, 20, 25\}$.

The upper panel of Figure 8 shows the average per-word perplexity over epochs with different number of topics. It is observed that in general, increasing topic numbers results in lower perplexity values. dJSTs trained with features *Filtered by POS* or augmented with nominal phrases (*Unigrams+phrases*) give lower perplexities than the original feature representation (*Filtered by stopword list*).

We also plot the average document-level sentiment classification accuracy over epochs with different number of topics as shown in the lower panel of Figure 8. It can be observed that models trained with features *Filtered by POS* outperform *Filtered by stopword list* under most topic settings. Augmenting the original bag-of-words feature space with nominal phrases (*Unigrams+phrases*) further improves the classification accuracy for both the *skip model* and *multiscale model*.

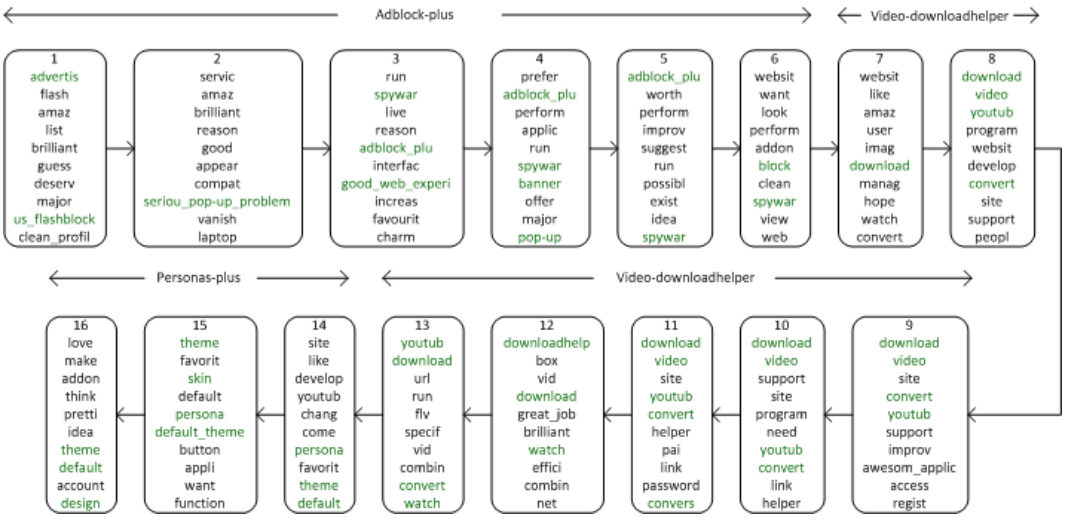
5.4. Example Topics

We list in Figure 9 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 10, 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 $P(z, l) = \frac{1}{|D_t|} \sum_{d \in D_t} P(z|l, d)P(l|d)$.

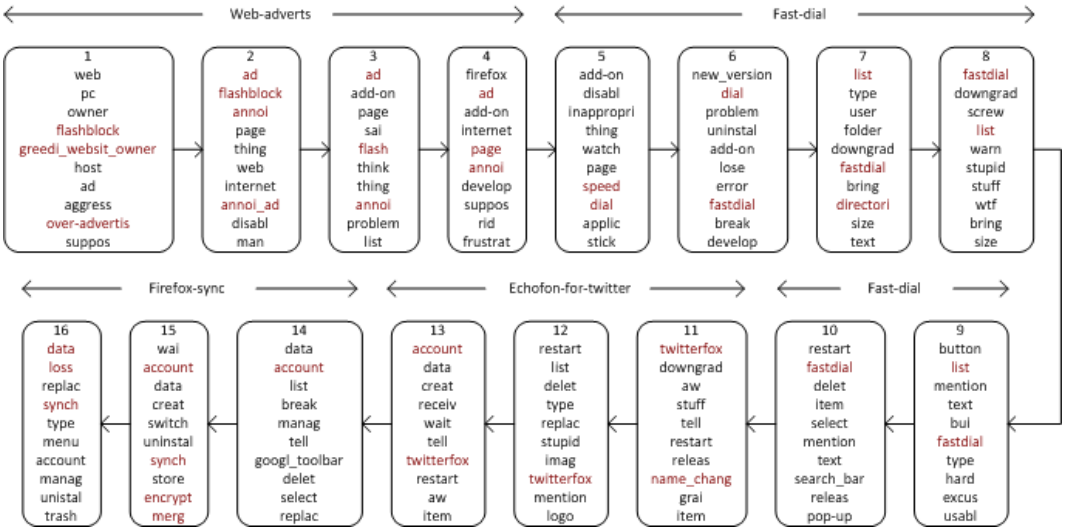
It was found in Figure 9 that the topics extracted from the input features comprising both unigrams and phrases are generally more meaningful than those from the bag-of-words representations, as phases such as '*good_web_experience*' and '*annoi_ad*' can deliver richer information. We also notice that the negative phrase '*seriou_pop-up_problem*' appears in the positive topic at Epoch 2. A manual examination on the original review text reveals that it actually appeared in a positive review about Adblock Plus with a user rating of 5 stars, *"...It's amazing! It even protected me on a graphics site that had got a serious pop-up problem. It's a must have. ..."*.

Figure 9 shows that the positive sentiment topics are mainly dominated by topics about Adblock Plus and Video DownloadHelper, with only the topics from the last three epochs mentioning Persona Plus. This observation is inline with the dataset statistics shown in Figure 3 that only reviews on Adblock Plus and Video DownloadHelper receive an average user rating of over 4.5 stars over the entire epoch history. Figure 10 also shows the prominence of positive sentiment topics about Adblock Plus in the first five epochs.

On the other hand, more topic transitions are observed for the negative sentiment topics, i.e., beginning with complains about web adverts, and then transits to negative comments about Fast Dial. At Epoch 8, there were a significantly high volume of reviews about Fast Dial and the average rating is about 2 stars. Hence, the negative sentiment topics about Fast Dial centered around Epoch 8. Negative topic transits to Echofon for Twitter at epoch 11 and to Firefox Sync at Epoch 14. Such a phenomenon can also be observed in Figure 10 that after Epoch 13, negative sentiment topics become more prominent than positive sentiment topics. This is consistent to what we



(a) Positive topics.



(b) Negative topics.

Fig. 9. Example topics evolved over time. Topic labels were derived from colour words and the number denotes epoch ID. Topics in upper and lower panels are the positive and negative sentiment topics respectively.

have observed in Figure 3 that there were an increasing number of reviews about Firefox Sync after Epoch 13 and the average user rating of Firefox Sync is only 2 stars.

6. 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

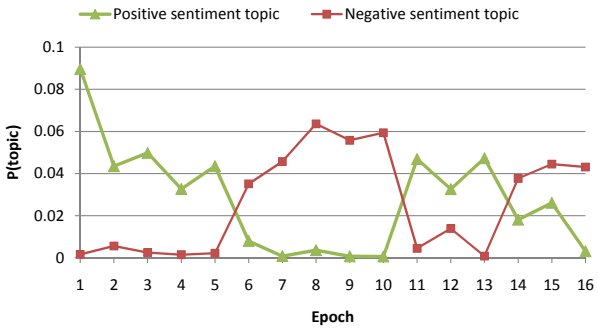


Fig. 10. Occurrence probability of topics with time. Positive and negative sentiment topics correspond to the topics listed in the upper and lower panel of Figure 9 respectively.

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 slightly better sentiment classification results than sliding window. In future work, we plan to evaluate the model in other social media domains such as Twitter and further investigate the model for large-scale data processing.

APPENDIX

This appendix shows the derivation of the estimation of the weight vector $\boldsymbol{\mu}^t$ of the dJST Model.

A. ESTIMATING THE WEIGHT VECTOR μ^T OF THE DJST MODEL

The weight vector $\boldsymbol{\mu}^t$ is estimated by maximizing the joint distribution of dJST using the fixed-point iteration method described in [Minka 2003]. We only need to focused the third term on the RHS of the joint distribution (Equation 2) as it is the only term that contains $\boldsymbol{\mu}^t$:

$$P(\boldsymbol{W}^t|\boldsymbol{L}^t,\boldsymbol{z}^t,\boldsymbol{E}^{t-1},\boldsymbol{\mu}^t)=\prod_{l=1}^{L}\prod_{z=1}^{T}\frac{\Gamma(\sum_s \mu_{l,z,s}^t)}{\prod_{w=1}^{V}\Gamma(\sum_s \mu_{l,z,s}^t\sigma_{l,z,s,w}^{t-1})}\frac{\prod_{w=1}^{V}\Gamma(N_{l,z,w}^t+\sum_s \mu_{l,z,s}^t\sigma_{l,z,s,w}^{t-1})}{\Gamma(N_{l,z}^t+\sum_s \mu_{l,z,s}^t)}. \tag{20}$$

Taking the log likelihood gives:

$$\log P(\boldsymbol{W}^t|\boldsymbol{L}^t,\boldsymbol{z}^t,\boldsymbol{E}^{t-1},\boldsymbol{\mu}^t)=\sum_{l=1}^{L}\sum_{z=1}^{T}\underbrace{[\log\Gamma(\sum_{z=1}^{T}\mu_{l,z,s}^t)-\log\Gamma(N_{l,z}^t+\sum_s \mu_{l,z,s}^t)]}_{T1}+$$
$$\sum_{l=1}^{L}\sum_{z=1}^{T}\sum_{w=1}^{V}\underbrace{[\log\Gamma(N_{l,z,w}^t+\sum_{s=1}^{S}\mu_{l,z,s}^t\sigma_{l,z,s,w}^{t-1})-\log\Gamma(\sum_{s=1}^{S}\mu_{l,z,s}^t\sigma_{l,z,s,w}^{t-1})]}_{T2}. \tag{21}$$

Terms T1 and T2 in Equation 21 can be bounded using the following bounds [Wallach 2008]

$$\log\Gamma(z) - \log\Gamma(z+n) \geq \log\Gamma(\hat{z}) - \log\Gamma(\hat{z}+n) + [\Psi(\hat{z}+n) - \Psi(\hat{z})](\hat{z}-z), \quad (22)$$

$$\log\Gamma(z+n) - \log\Gamma(z) \geq \log\Gamma(\hat{z}+n) - \log\Gamma(\hat{z}) + \hat{z}[\Psi(\hat{z}+n) - \Psi(\hat{z})](\log z - \log\hat{z}). \quad (23)$$

Applying bounds 22 and 23 to Equation 21 yields

$$\begin{aligned}
\log P(\boldsymbol{W}^t|\boldsymbol{L}^t, \boldsymbol{z}^t, \boldsymbol{E}^{t-1}, \boldsymbol{\mu}^t) \geq & \sum_{l=1}^{L}\sum_{z=1}^{T}\{\log\Gamma(\sum_{s'=1}^{S}\mu_{l,z,s'}^t) - \log\Gamma(N_{l,z}^t + \sum_{s'=1}^{S}\mu_{l,z,s'}^t) + \\
& [\Psi(N_{l,z}^t + \sum_{s'=1}^{S}\mu_{l,z,s'}^t) - \Psi(\sum_{s'=1}^{S}\mu_{l,z,s'}^t)](\sum_{s'=1}^{S}\mu_{l,z,s'}^S - \sum_{s=1}^{S}\mu_{l,z,s}^t)\} + \\
& \sum_{l=1}^{L}\sum_{z=1}^{T}\sum_{w=1}^{V}\{\log\Gamma(N_{l,z,w}^t + \sum_{s'=1}^{S}\mu_{l,z,s'}\sigma_{l,z,s',w}^{t-1}) - \log\Gamma(\sum_{s'=1}^{S}\mu_{l,z,s'}\sigma_{l,z,s',w}^{t-1}) + \\
& \sum_{s'=1}^{S}\mu_{l,z,s'}\sigma_{l,z,s',w}^{t-1}[\,\Psi(N_{l,z,w}^t + \sum_{s'=1}^{S}\mu_{l,z,s'}^t\sigma_{l,z,s',w}^{t-1}) - \Psi(\sum_{s'=1}^{S}\mu_{l,z,s'}^t\sigma_{l,z,s',w}^{t-1})\,]\times \\
& \underbrace{[\,\log(\sum_{s=1}^{S}\mu_{l,z,s}^t\sigma_{l,z,s,w}^{t-1})}_{T3} - \log(\sum_{s'=1}^{S}\mu_{l,z,s'}^t\sigma_{l,z,s',w}^{t-1})\,]\,\},
\end{aligned} \quad (24)$$

where term T3 in Equation 24 can be further bounded using the following bound

$$\log(a+b) \geq \log a + \log b, \quad (25)$$

giving

$$\log(\sum_{s=1}^{S}\mu_{l,z,s}^t\sigma_{l,z,s,w}^{t-1}) \geq \sum_{s=1}^{S}(\log\mu_{l,z,s}^t + \log\sigma_{l,z,s,w}^{t-1}). \quad (26)$$

Differentiating Equation 24 with respect to $\mu_{l,z,s}$ gives:

$$\begin{aligned}
\frac{\partial \log P(\boldsymbol{W}^t|\boldsymbol{L}^t, \boldsymbol{z}^t, \boldsymbol{E}^{t-1}, \boldsymbol{\mu}^t)}{\partial \mu_{l,z,s}^t} \geq & -\underbrace{[\,\Psi(N_{l,z}^t + \sum_{s'=1}^{S}\mu_{l,z,s'}^t) - \Psi(\sum_{s'=1}^{S}\mu_{l,z,s'})\,]}_{B_{l,z}^t} + \\
& \sum_{w=1}^{V}\sum_{s'=1}^{S}\mu_{l,z,s'}^t\sigma_{l,z,s',w}^{t-1}\underbrace{[\,\Psi(N_{l,z,w}^t + \sum_{s'=1}^{S}\mu_{l,z,s'}^t\sigma_{l,z,s',w}^{t-1}) - \Psi(\sum_{s'=1}^{S}\mu_{l,z,s'}^t\sigma_{l,z,s',w}^{t-1}))\,]}_{A_{l,z,w}^t}\frac{1}{\mu_{l,z,s}}
\end{aligned} \quad (27)$$

Setting the differentiation to 0 gives:

$$(\mu_{l,z,s}^t)^{new} = \frac{\mu_{l,z,s'}\sum_{w=1}^{V}\sigma_{l,z,s',w}^{t-1}\cdot A_{l,z,w}^t}{B_{l,z}^t} \quad (28)$$

ACKNOWLEDGMENTS

The authors would like to thank Dong Liu for crawling Mozilla add-ons review data and Stefan Geissler for providing the part-of-speech tagging and syntactic parsing results.

REFERENCES

AHMED, A. AND XING, E. 2008. Dynamic non-parametric mixture models and the recurrent chinese restaurant process. In *SDM*.

BLEI, D. AND LAFFERTY, J. 2006. Dynamic topic models. In *ICML*. 113–120.

BOLLEN, J., MAO, H., AND PEPE, A. 2010a. Determining the public mood state by analysis of microblogging posts. In *Proceedings of the Alife XII Conference*.

BOLLEN, J., PEPE, A., AND MAO, H. 2010b. *Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena*. http://arxiv.org/abs/0911.1583.

CHAKRABARTI, D., KUMAR, R., AND TOMKINS, A. 2006. Evolutionary clustering. In *KDD*. 554–560.

CHI, Y., SONG, X., HINO, K., AND TSENG, B. 2007. Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD*. 153–162.

CHI, Y., SONG, X., ZHOU, D., HINO, K., AND TSENG, B. L. 2009. On evolutionary spectral clustering. *ACM Transactions on Knowledge Discovery from Data 3*, 17:1–17:30.

HE, Y. AND LIN, C. 2012. Online Sentiment and Topic Dynamics Tracking over the Streaming Data. In *Proceedings of the ASE/IEEE International Conference on Social Computing (SocialCom)*. Amsterdam, The Netherlands.

IWATA, T., YAMADA, T., SAKURAI, Y., AND UEDA, N. 2010. Online multiscale dynamic topic models. In *KDD*. 663–672.

LIN, C. AND HE, Y. 2009. Joint sentiment/topic model for sentiment analysis. In *CIKM*. 375–384.

LIN, C., HE, Y., EVERSON, R., AND RUEGER, S. 2012. Weakly-supervised Joint Sentiment-Topic Detection from Text. *IEEE Transactions on Knowledge and Data Engineering 24,* 6, 1134–1145.

MAO, Y. AND LEBANON, G. 2007. Isotonic conditional random fields and local sentiment flow. In *NIPS*. Vol. 19. 961–968.

MAO, Y. AND LEBANON, G. 2009. Generalized isotonic conditional random fields. *Machine learning 77,* 2, 225–248.

MCNAIR, D., LORR, M., AND DROPPLEMAN, L. 1992. *Profile of Mood States: POMS*. EdiTS, Educational and Industrial Testing Service.

MEI, Q., LING, X., WONDRA, M., SU, H., AND ZHAI, C. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*. 171–180.

MEI, Q. AND ZHAI, C. 2005. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD*. 198–207.

MINKA, T. 2003. Estimating a Dirichlet distribution. Tech. rep.

NALLAPATI, R., DITMORE, S., LAFFERTY, J., AND UNG, K. 2007. Multiscale topic tomography. In *KDD*. 520–529.

O'CONNOR, B., BALASUBRAMANYAN, R., ROUTLEDGE, B., AND SMITH, N. 2010. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*. 122–129.

PRUTEANU-MALINICI, I., REN, L., PAISLEY, J., WANG, E., AND CARIN, L. 2009. Hierarchical Bayesian modeling of topics in time-stamped documents. *IEEE transactions on pattern analysis and machine intelligence*.

REN, L., DUNSON, D., AND CARIN, L. 2008. The dynamic hierarchical Dirichlet process. In *ICML*. 824–831.

STEYVERS, M. AND GRIFFITHS, T. 2007. Probabilistic Topic Models. *Handbook of Latent Semantic Analysis*, 427–446.

TEH, Y., JORDAN, M., BEAL, M., AND BLEI, D. 2006. Hierarchical dirichlet processes. *Journal of the American Statistical Association 101,* 476, 1566–1581.

WALLACH, H. 2008. Structured topic models for language. Ph.D. thesis, University of Cambridge.

WALLACH, H., MIMNO, D., AND MCCALLUM, A. 2009. Rethinking lda: Why priors matter. *Advances in Neural Information Processing Systems 22*, 1973–1981.

WANG, C., BLEI, D., AND HECKERMAN, D. 2008. Continuous time dynamic topic models. In *Proc. of UAI*.

WANG, X. AND MCCALLUM, A. 2006. Topics over time: a non-Markov continuous-time model of topical trends. In *KDD*. 424–433.

XU, T., ZHANG, Z., YU, P., AND LONG, B. 2008a. Dirichlet process based evolutionary clustering. In *ICDM*. 648–657.

XU, T., ZHANG, Z., YU, P., AND LONG, B. 2008b. Evolutionary clustering by hierarchical dirichlet process with hidden markov state. In *ICDM*. 658–667.

ZHANG, J., SONG, Y., ZHANG, C., AND LIU, S. 2010. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *KDD*. 1079–1088.

Received February 2012; revised XXX 2013; accepted XXX 2013