University of Exeter
Department of Computer Science

Probabilistic topic models for sentiment analysis on the Web

Chenghua Lin

September 2011

Submitted by Chenghua Lin, to the the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science , September 2011.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other University.

(signature) ....

Dedicated to my parents, Xingwen Lin and Yuanfang Chen.

Acknowledgements

This thesis would not have been possible without the help and support from the following people.

First and foremost, I would like to thank my supervisor Yulan He, who has been an exemplary teacher and mentor in the past three years of my PhD studies. I am extremely grateful to Yulan's encouragement, support and patience during the difficult time of my research.

My gratitude also goes to my second supervisor Richard Everson, whose input and advice on my work were very useful. Richard is also very generous with his time for discussing research, for which I am grateful.

I also thank Alex Gong, without whose kind help I would not have been able to pursue the research I really enjoy.

I am fortunate to have a number of great friends who have supported me both research wise and non-research wise, particularly Naihui He, Junaid Khan, and Bing Qiao. Without them, my PhD would have been a less enjoyable one.

Deepest thanks to my parents and family. Their lifetime love and care help me regain my confidence and make me once again believe in myself. Finally, I thank Yuan. Her kindness, patience and care sustain me.

Abstract

Sentiment analysis aims to use automated tools to detect subjective information such as opinions, attitudes, and feelings expressed in text, and has received a rapid growth of interest in natural language processing in recent years. Probabilistic topic models, on the other hand, are capable of discovering hidden thematic structure in large archives of documents, and have been an active research area in the field of information retrieval. The work in this thesis focuses on developing topic models for automatic sentiment analysis of web data, by combining the ideas from both research domains.

One noticeable issue of most previous work in sentiment analysis is that the trained classifier is domain dependent, and the labelled corpora required for training could be difficult to acquire in real world applications. Another issue is that the dependencies between sentiment/subjectivity and topics are not taken into consideration. The main contribution of this thesis is therefore the introduction of three probabilistic topic models, which address the above concerns by modelling sentiment/subjectivity and topic simultaneously.

The first model is called the joint sentiment-topic (JST) model based on latent Dirichlet allocation (LDA), which detects sentiment and topic simultaneously from text. Unlike supervised approaches to sentiment classification which often fail to produce satisfactory performance when applied to new domains, the weakly-supervised nature of JST makes it highly portable to other domains, where the only supervision information required is a domain-independent sentiment lexicon. Apart from document-level sentiment classification results, JST can also extract sentiment-bearing topics automatically, which is a distinct feature compared to the existing sentiment analysis approaches.

The second model is a dynamic version of JST called the dynamic joint sentiment-topic (dJST) model. dJST respects the ordering of documents, and allows the analysis of topic and sentiment evolution of document archives that are collected over a long time span. By accounting for the historical dependencies of documents from the past epochs in the generative process, dJST gives a richer posterior topical structure than JST, and can better respond to the permutations of topic prominence. We also derive online inference procedures based on a stochastic EM algorithm for efficiently updating the model parameters.

The third model is called the subjectivity detection LDA (subjLDA) model for sentence-level subjectivity detection. Two sets of latent variables were introduced in subjLDA. One is the subjectivity label for each sentence; another is the sentiment label for each word token. By viewing the subjectivity detection problem as weakly-supervised generative model learning, subjLDA significantly outperforms the baseline and is comparable to the supervised approach which relies on much larger amounts of data for training.

These models have been evaluated on real world datasets, demonstrating that joint sentiment topic modelling is indeed an important and useful research area with much to offer in the way of good results.

Publications

Chapter 3 is based on the previously published work:

> Lin, C., He, Y., Everson, R. and Rüger, S. Weakly-supervised Joint Sentiment-Topic Detection from Text, *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, to appear.
>
> Lin, C., He, Y., and Everson, R. A Comparative Study of Bayesian Models for Unsupervised Sentiment Detection, In *Proceedings of the 14th Conference on Computational Natural Language Learning (CoNLL)*, Uppsala, Sweden, 2010.
>
> Lin, C. and He, Y. Joint Sentiment/Topic Model for Sentiment Analysis, In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM)*, Hong Kong, China, 2009.

Chapter 5 is based on the previously published work:

> Lin, C., He, Y. and Everson, R. Sentence Subjectivity Detection with Weakly-Supervised Learning, In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, Chiang Mai, Thailand, 2011.

Other publications:

> He, Y., Lin, C. and Alani, H. Automatically Extracting Polarity-Bearing Topics for Cross-Domain Sentiment Classification, In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, Portland, Oregon, 2011.
>
> He, Y. and Lin, C. Protein-Protein Interactions Classification from Text via Local Learning with Class Priors, In *Proceedings of the 14th International Conference on Applications of Natural Language to Information Systems (NLDB)*, Saabrucken, Germany, 2009.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent surveys have revealed that opinion-rich resources such as online reviews and social networks are having greater economic impact on both consumers and companies compared to traditional media [Pang and Lee, 2008]. Driven by the demand of gleaning insights into such vast amount of user-generated data, work on developing new algorithms for automated sentiment analysis has bloomed in the past few years. On the other hand, probabilistic topic models [Blei et al., 2003] have attracted a dramatic surge of interest in the field of information retrieval, owing to their capability of discovering the hidden thematic structures in large archives of documents. The work in this thesis therefore focuses on developing topic models for automatic sentiment analysis of web data, by combining the ideas from both research domains described above.

1.1 Sentiment Analysis

Sentiment analysis aims to use automated tools to detect subjective information such as opinions, attitudes, and feelings in text expressed by the opinion holder. This thesis work focuses on two main sentiment analysis tasks here, sentiment classification and subjectivity detection.

Sentiment Classification In sentiment classification, one tries to predict whether the sentiment orientation of a given text is positive, negative or neutral. When classification involves only two classes of labels (i.e. either positive or negative), it is also known as polarity classification [Pang et al., 2002]. Research on sentiment classification has attracted a great deal of attention, where different classification tasks focus on various levels of granularity, e.g., from the document level [Pang et al., 2002], to the finer-grained sentence [Kim and Hovy, 2004] and word/phrase level [Turney and Littman, 2002]. Apart from the research challenges, the growing interest in this area is largely due to the many useful applications it can offer, such as predicting stock market behaviour based on the sentiment results of Twitter posts [Bollen et al., 2011], measuring public poll opinion of presidential elections from Blog data [OConnor et al., 2010], and handling business intelligence tasks related to customer feedback [Pang and Lee, 2008].

Subjectivity Detection In contrast to sentiment classification that seeks to identify the sentiment orientation of a given opinionated text, subjectivity detection classifies whether the given text expresses opinions (subjective) or reports facts (objective). Such a task of distinguishing subjective information from objective has been reported to be more difficult than sentiment classification [Mihalcea et al., 2007], and is useful for many natural language processing applications. For instance, it is often assumed in sentiment classification that the input documents are opinionated, and ideally contain subjective statements only [Yu and Hatzivassiloglou, 2003]; document summarization systems need to summarize different perspectives and opinions [Pang and Lee, 2008]; for question answering systems, extracting and presenting information of the appropriate type (i.e. opinions or facts) is imperative according to the specific question being asked [Wiebe and Riloff, 2005].

1.2 Probabilistic Topic Models

Probabilistic topic models such as latent Dirichlet allocation (LDA) [Blei et al., 2003] and probabilistic latent semantic analysis (PLSA) [Hofmann, 1999] can capture the semantic properties of documents by modelling documents as a mixture of distributions over words, which are known as topics. The words under each topic tend to co-occur in the same document, and generally have a tight semantic relation with one another. So by inspecting the topics that have high probabilities associating with a document, one can easily interpret what that document is about. For instance, restaurant reviews probably discuss topics about food, service, location, price, etc; whereas electronic device reviews probably cover aspects such as design, quality and functions. Another advantage of topic models is that model learning is fully unsupervised, which does not require any prior annotation or labelling of documents. Hence, topic modelling is suitable for managing and summarizing large document archives [Blei, 2011], and has seen many successful applications such as in analysis of scientific journals [Blei and Lafferty, 2006], Wikipedia articles [Hoffman et al., 2010], academic email [McCallum et al., 2007], and navigation tools for digital libraries [Mimno and McCallum, 2007], etc.

While the existing sentiment analysis approaches [Bollen et al., 2011; Kim and Hovy, 2004; OConnor et al., 2010; Pang et al., 2002; Turney and Littman, 2002; Yu and Hatzivassiloglou, 2003] have enabled various types of intelligence applications, they mainly focused on detecting the overall sentiment or subjectivity of text without considering the topical information, which makes the results less informative to users. Topic models, on the other hand, provide means for organizing and summarizing large digitized archives of information in an unsupervised fashion [Blei et al., 2003; Griffiths and Steyvers, 2004]. However, work on this line detects topics alone, with the correlations between topic and sentiment untouched.

The work in this thesis is illuminated by the two research areas described above, to develop new probabilistic topic models for sentiment analysis of web data. Specifically, these new models meet the needs of analysing large collections of opinionated documents, by addressing the limitations of current sentiment analysis approaches which are identified in the following section.

1.3 Research Challenges

In this section, several research challenges in the field of sentiment analysis are identified. These intellectual challenges arise from the tasks of both sentiment classification and subjectivity detection, which motivate the development of the new models in the thesis.

1.3.1 Training Sentiment Classifier without Labelled Data

Among various computational treatments to sentiment analysis, a large portion of work concentrates on classifying a sentiment-bearing document according to its sentiment polarity, i.e. either positive or negative as a binary classification. Although much work has been done in this line [Kennedy and Inkpen, 2006; Pang and Lee, 2004; Pang et al., 2002; Turney, 2001; Whitelaw et al., 2005], most of the existing approaches rely on supervised learning models trained from labelled corpora where each document has been labelled as positive or negative prior to training. However, such labelled corpora are not always easy to obtain in practical applications. Additionally, it is well-known that sentiment classifiers trained on one domain often fail to produce satisfactory results when applied to other domains [Aue and Gamon, 2005; Blitzer et al., 2007]. For example, it was reported by Aue and Gamon [2005] that an in-domain Support Vector Machine (SVM) classifier trained on movie review data (giving best accuracy of 90.45%) achieved relatively poor accuracies of 70.29% and 61.36% respectively when tested on book review and product support services data. Such phenomena reflect the fact that sentiments are context dependent, so that sentiment expressions can be quite different for different topics or domains. For instance, when appearing under different topics within movie review data, the adjective "*complicated*" may have negative sentiment orientation as "*complicated role*" in one topic, and conveys positive sentiment as "*complicated plot*" in another topic. This suggests that modelling sentiment and topic simultaneously may help find better feature representations for sentiment classification. Therefore, the first research challenge is to develop unsupervised or weakly-supervised sentiment models that do not require labelled data for training, and yet account for the topic or domain dependencies in sentiment classification.

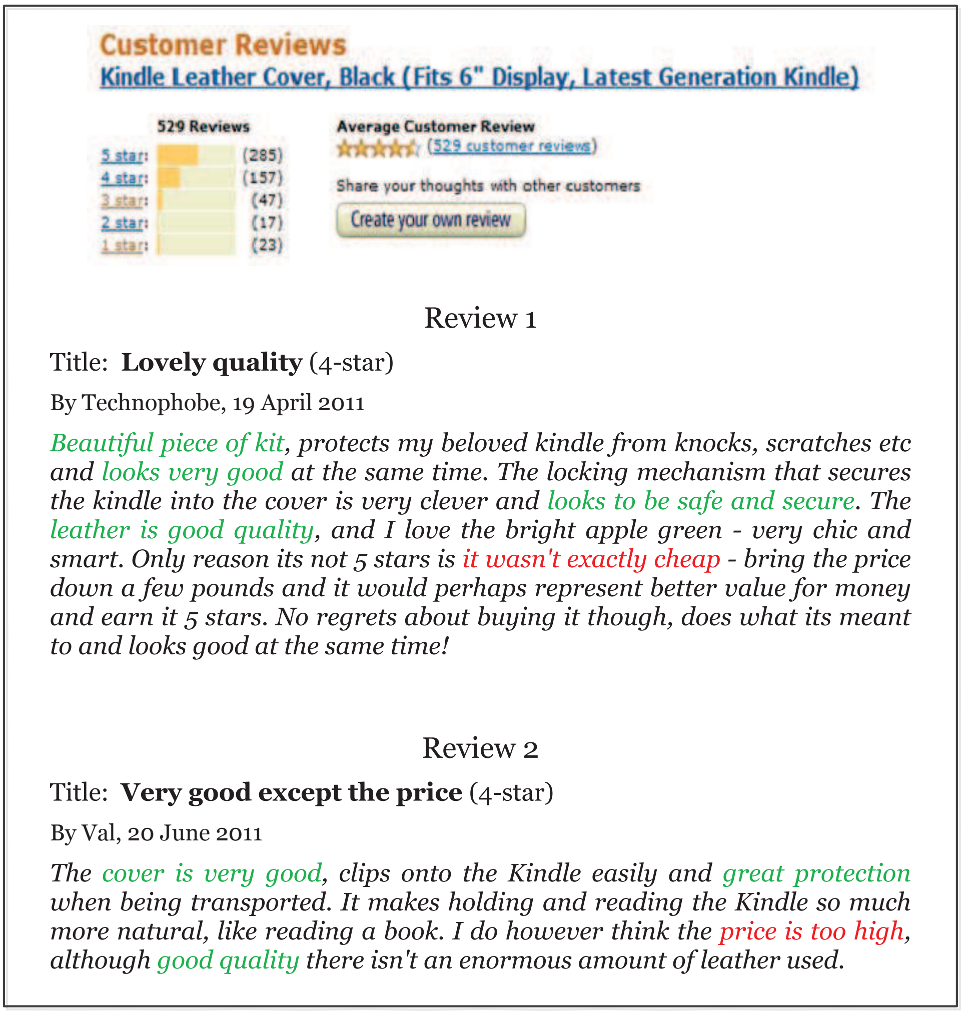
1.3.2 Simultaneously Detecting Topic and Sentiment

From the application perspective, although it is useful to detect the overall sentiment of a document [Kennedy and Inkpen, 2006; Pang and Lee, 2004; Pang et al., 2002; Whitelaw et al., 2005], it is just as useful, and perhaps even more interesting, to understand the underlying topics and the associated topic sentiments of a document. Take the Amazon Kindle cover reviews shown in Figure 1.1 as an example. This Kindle cover receives a very high average rating from a total of 529 reviews. However, the review bar-chart shows that there are still quite a lot of customers who only gave a 4- or 3-star rating to the product.

When making a purchase, despite the overall rating, it would be very helpful to know the pros and cons of the product being discussed in the reviews. An inspection of those two hundred 4- and 3- star reviews reveals that actually, many people think the cover *design* and *quality* are very good, but it is just *overpriced*. Having obtained such information, the Amazon Kindle cover would still be the best buy for the customers with large budgets, while others may choose a less expensive alternative. Nevertheless, people can still easily be overwhelmed by the quantity and variety of the available data, such as the example given above. So the second challenge is to find solutions that can automatically extract sentiment-bearing statements relating to product aspects in order to alleviate the information seeking burden of users.

1.3.3 Capturing Sentiment and Topic Dynamics

Many document collections exhibit dynamics, where the patterns of the documents collected at an earlier stage may not be preserved subsequently, especially for collections



Customer Reviews

Kindle Leather Cover, Black (Fits 6" Display, Latest Generation Kindle)

529 Reviews

5 star:	(285)
4 star:	(157)
3 star:	(47)
2 star:	(17)
1 star:	(23)

Average Customer Review

(529 customer reviews)

Share your thoughts with other customers

Review 1

Title: **Lovely quality** (4-star)

By Technophobe, 19 April 2011

Beautiful piece of kit, protects my beloved kindle from knocks, scratches etc and looks very good at the same time. The locking mechanism that secures the kindle into the cover is very clever and looks to be safe and secure. The leather is good quality, and I love the bright apple green - very chic and smart. Only reason its not 5 stars is it wasn't exactly cheap - bring the price down a few pounds and it would perhaps represent better value for money and earn it 5 stars. No regrets about buying it though, does what its meant to and looks good at the same time!

Review 2

Title: **Very good except the price** (4-star)

By Val, 20 June 2011

The cover is very good, clips onto the Kindle easily and great protection when being transported. It makes holding and reading the Kindle so much more natural, like reading a book. I do however think the price is too high, although good quality there isn't an enormous amount of leather used.

Figure 1.1: Amazon Kindle cover reviews. Text highlighted in green and red indicate the pros and cons of the product respectively.

that span years or decades. For instance, a mobile application which receives very positive reviews after the release of a first version, could a few months later, as a newer version introduces a bug, receive more prominent negative reviews. However, when analysing a document collection, topic models like LDA posit an assumption that documents have static word co-occurrence patterns, and that the order of documents does not matter. This assumption may not be realistic for many real world datasets such as the mobile review example. For these time-variant collections, we may want to assume that topics change over time, as well as the sentiments associated with the topics. Thus, it is necessary to develop dynamic models that can detect and track the shifts in both topic and sentiment over time in time-variant datasets, which is the third challenge.

1.3.4 Subjectivity Detection without Labelled Data

Work on sentence-level subjectivity detection is relatively sparse compared to document-level sentiment classification. Early work used a bootstrapping algorithm to learn subjective [Riloff and Wiebe, 2003] or both subjective and objective [Wiebe and Riloff, 2005] expressions for sentence-level subjectivity detection. In contrast to bootstrapping, there have been some recent attempts exploring various n-gram features and different level of lexical instantiation for detecting subjective utterance from conversation data [Murray and Carenini, 2009; Raaijmakers et al., 2008; Wilson and Raaijmakers, 2008]. However, the aforementioned line of work tackled subjectivity detection either as supervised or semi-supervised learning, requiring labelled data and extensive knowledge for training, despite the fact that the gold standard labels at the sentence level could be prohibitively expensive to acquire. Additionally, similar to sentiment classification, subjectivity is also context sensitive, so that subjectivity classifiers trained on one domain often suffer from performance loss when applied to new domains [Pang and Lee, 2008]. Motivated by the above observations, the final challenge of the thesis is to develop a subjectivity detection algorithm that is relatively simple compared to the existing methods (e.g. based on bootstrapping or n-gram features), and yet can easily be transferred between domains through unsupervised or weakly-supervised learning without using any labelled data.

1.4 Thesis Contribution

In this thesis, three new topic models are proposed to address the research challenges in sentiment analysis described in Section 1.3. The contributions of this thesis can be summarized in the following aspects:

- A novel probabilistic modeling framework called the joint sentiment-topic (JST) model based on latent Dirichlet allocation (LDA), which detects sentiment and topic simultaneously from text. A mechanism is introduced to incorporate prior information about the sentiment lexicons into model learning by modifying the Dirichlet priors of the topic-word distributions. Unlike supervised approaches to sentiment classification, which often fail to produce satisfactory performance when applied to other domains, the weakly-supervised nature of JST makes it highly portable to other domains, as will be verified by the experimental results on datasets from five different domains. Moreover, the topics and topic sentiment detected by JST are indeed coherent and informative.

- A new member of the probabilistic time series models called dynamic joint sentiment-topic (dJST) model, for sequentially analysing the topic and sentiment evolution over time in a document collection. The dJST model respects the ordering of documents. By accounting for the historical dependencies of documents from the past epochs in the model generative process, dJST gives a richer posterior topical structure than JST, and thus can better respond to the permutations of topic prominence. For efficiently analysing large corpora, we derived online inference procedures based on a stochastic EM algorithm, allowing the dJST model to be updated sequentially using the newly arrived data and the parameters of the previously estimated model.
- A novel hierarchical topic model called the subjectivity detection LDA (subjLDA) model, for sentence-level subjectivity detection. The only supervision information needed for subjLDA is a small set of domain-independent subjectivity clues; no labelled documents are used. By viewing the subjectivity detection problem as weakly-supervised generative model learning, subjLDA significantly outperforms the baseline and is comparable to models which rely on a much larger amount of data for training [Wiebe and Riloff, 2005].

1.5 Thesis Overview

This chapter has briefly introduced the research area of sentiment analysis and identified a few limitations of the current sentiment analysis approaches. It then summarized the main contributions of the thesis work in response to the research challenges identified. The remainder of the thesis is structured as follows:

Chapter 2 starts with a literature review of the work on sentiment analysis, which is categorized into supervised, semi-supervised and unsupervised learning. We then review the topic models on which the thesis work is based. The advantages and disadvantages of various approaches are also compared.

Chapter 3 presents the joint sentiment-topic (JST) model and a reparameterized version of JST called the Reverse-JST model. The mathematical framework of JST and a Gibbs sampling algorithm for estimating the model parameters are described in detail. Experimental evaluation using the movie review (MR)[1] and the multi-domain sentiment (MDS)[2] datasets is also presented.

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data

[2] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/index2.html

Chapter 4 introduces the dynamic joint sentiment-topic (dJST) model by extending the JST model framework. An efficient stochastic EM algorithm for online updating the model parameters with the historical information and the newly arrived data is described in detail. In the experimental evaluation, the dJST model performance is compared to the non-dynamic JST and LDA models using the Mozilla add-on review dataset which was collected as part of the PhD study.

Chapter 5 introduces the subjectivity detection LDA (subjLDA) model for sentence-level subjectivity detection. The subjLDA model framework is first described, followed by the model inference using the Gibbs sampling algorithm. In the evaluation, the subjLDA model performance is compared to the baseline and the LDA model, based on the Multi-Perspective Question Answering (MPQA)[1] dataset.

Chapter 6 summaries the thesis work and outlines a few possible directions for future research.

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

Chapter 2

Related Work

This thesis work focuses on developing topic models for automatic sentiment analysis. Therefore, we first survey a range of different machine learning techniques for sentiment analysis in Section 2.1, and then in Section 2.2, we discuss related work on topic models which form the ground of the models we have developed in this thesis.

2.1 Sentiment Analysis

Sentiment classification seeks to identify the polarity of texts according to the sentiment expressed by the opinion holder. In this section, we investigate the work which deals with computational treatments of sentiment using different machine learning techniques, with a focus on document-level sentiment classification.

2.1.1 Supervised Approaches

Most supervised sentiment classification approaches use standard machine learning techniques such as support vector machines (SVMs) and Naive Bayes (NB) classifiers. These approaches are corpus-based, in which a domain-specific classifier is trained with labelled training data.

Pioneering work on document-level sentiment classification is by Pang et al. [2002], who employed machine learning techniques including SVMs, NB and Maximum Entropy (MaxEnt) to determine whether the sentiment expressed in a movie review was "*thumbs up*" or "*thumbs down*". They achieved the best classification accuracy with SVMs using binary features coding whether a unigram was present or not. In subsequent work, Pang and Lee

[2004] further improved sentiment classification accuracy on the movie review dataset using a cascaded approach. Instead of training a classifier on the original feature space, they first filtered out the objective sentences from the dataset using a global min-cut inference algorithm, and then used the remaining subjective sentences as input for sentiment classifier training. The classification improvement achieved by the cascaded approach suggests that the subjective sentences contain features which are more discriminative and informative than the full dataset for sentiment classification. The movie review dataset (also known as the polarity dataset) used in [Pang and Lee, 2004; Pang et al., 2002] has later on become a benchmark for many sentiment classification studies [Matsumoto et al., 2005; Whitelaw et al., 2005]. Whitelaw et al. [2005] used fine-grained semantic distinctions in features for sentiment classification, namely the appraisal groups. Specifically, a *appraisal group* is defined as coherent groups of words that express together a particular attitude, such as "*extremely boring*" and "*not terribly funny*". By training a SVM classifier on the combination of different types of appraisal group features and bag-of-word features, they achieved the best accuracy of 90.2% on the movie review dataset. Matsumoto et al. [2005] proposed a method using the extracted word sub-sequences and dependency sub-trees as features for SVMs training and attained the state-of-the-art accuracy of 93.7%.

A common assumption made by the aforementioned line of work [Pang and Lee, 2004; Pang et al., 2002; Whitelaw et al., 2005] is that the entire document is represented as a flat feature vector (i.e. a bag-of-words format), which limits their ability to exploit sentiment or subjectivity information at a finer-grained level. In this regard, there has been work on incorporating sentence or sub-sentence level sentiment label information for document-level sentiment classification [McDonald et al., 2007; Yessenalina et al., 2010; Zaidan et al., 2007].

McDonald et al. [2007] proposed a fully supervised structured model for joint sentence- and document-level sentiment classification based on sequence classification techniques using constrained Viterbi inference. The joint model leverages both document-level and sentence-level label information, and allows classification decisions from one level (e.g. the document level) to influence decisions at another level (e.g. the sentence level). It was reported that the joint model significantly outperformed both the document- and sentence-classifier that predict a single level label only. Zaidan et al. [2007] used human annotators to mark the sub-sentence level text spans known as "*annotator rationales*", which support the document's sentiment label. These annotator rationales were used as additional constraints for SVMs training, which ensure that the resulting classifier will be less confident in classifying the documents that do not contain the rationales. By exploiting the rationales during the classifier training, the proposed approach achieved 92.2% accuracy on the movie review dataset, and significantly outperformed the baseline SVM which only used the full text of the original documents for training.

More recently, Yessenalina et al. [2010] proposed a supervised multi-level structured model based on structural SVMs, which learns to jointly predict the document label and the labels of a sentence subset that best explain the document sentiment (i.e. the explanatory sentences). By treating the sentence-level labels as hidden variables, the proposed model does not required sentence-level annotation for training and thus avoids the lower-level labelling cost. Additionally, as opposed to the structured model [McDonald et al., 2007] which optimizes the accuracy for all levels being modelled, they took a different view to formulate the training objective to directly optimize for document-level accuracy. This multi-level structured model achieved 93.22% document-level sentiment classification accuracy on the movie review dataset, which outperformed both the structured model [McDonald et al., 2007] and the approach of Zaidan et al. [2007]. Nevertheless, a reasonable initial guess of the explanatory sentences is required in order to train the model properly.

2.1.2 Semi-supervised Approaches

Supervised sentiment classification approaches usually perform well when the training set is large enough. However, it is well known that the trained classifier often fail to produce satisfactory performance when the test data distribution is significantly different from the distribution of training data. As such, one would have to collect and label data for classifier retraining whenever a new domain is encountered. However, online content varies widely in domains and evolves rapidly over time, making corpora annotation for each domain unrealistic. In response to the labelling cost problem faced by supervised approaches in the sentiment classification task, there has been rising interest in exploring semi-supervised methods which leverage a large amount of unlabelled data and a small amount of labelled data for classifier training, especially for the tasks of domain adaptation and cross-lingual sentiment classification.

2.1.2.1 Monolingual Domain Adaptation

There has been a significant amount of work on domain adaptation for sentiment classification. Aue and Gamon [2005] explored various strategies for training SVM classifiers for the target domain lacking sufficient labelled data, where training data is obtained from other domains with rich labelled examples. These strategies include (1) training on a mixture of all the available labelled data (also used as baseline); (2) training on all the available labelled data, but limiting the set of features to those observed in the target domains; (3) using ensembles of classifiers from domains with available labelled data; and (4) combining small amounts of unlabelled data with large amounts of unlabelled data in the target domain for classifier training with an EM algorithm. It was found that the EM approach provided the best classification accuracy of the four strategies because it can take advantage of unlabelled data in the target domain for training. Tan et al. [2007] also leveraged data from both source and target domain for sentiment adaptation, where the target domain data are completely unlabelled. The main idea of their approach is to use classifiers trained on source domains to label some informative documents in the target domain. Those informative documents, picked up by a relative similarity ranking (RSR) method, were then used to retrain a centroid classifier for the target domain sentiment classification. The approach of Tan *et al.* outperformed the transductive SVM (TSVM) baseline classifier in most of the evaluation tasks, obtaining an average of more than 80% accuracy.

In a similar vein, Blitzer et al. [2007] addressed the domain transfer problem with the structural correspondence learning (SCL) algorithm using labelled data from a source domain and unlabelled data from both source and target domains. A key factor to the efficiency of SCL is the selection of pivot features, which are used to link the source and target domains. The pivot feature selection were performed in two stages. Frequent words in both source and target domains were first selected as candidate pivot features and pivots were then chosen based on the mutual information between these candidate features and the source labels. They reduced the relative error due to adaptation between domains by an average of 46% over a supervised baseline model (a linear classifier trained to minimize the Huber loss with stochastic gradient descent), based on the multi-domain sentiment (MDS) dataset[1]. In subsequent work, Li and Zong [2008] explored two methods for multi-domain sentiment classification, which are very similar to the approaches of Aue and Gamon [2005]. The first one combines the feature sets from all domains into one feature set for classifier training (feature fusion); the other combines multiple single classifiers trained on individual domains using meta-learning (classifier fusion). While both methods can achieve better results than the baseline which used single domain supervised learning, it was found that the classifier fusion method consistently outperformed the feature fusion method, which is in line with the observations of Aue and Gamon [2005].

[1] `http://www.cs.jhu.edu/~mdredze/datasets/sentiment/`

Instead of solely relying on source and target domain data for training, there have been studies which incorporate sentiment lexicons, heuristic knowledge, or augmenting feature spaces for improving sentiment classification accuracy. Li et al. [2009] proposed a non-negative matrix tri-factorization approach for semi-supervised sentiment classification, which learns from lexical prior knowledge of domain-independent sentiment lexicons in conjunction with domain-dependent unlabelled data and a few labelled documents. Although not directly targeting the domain adaptation problem, the proposed approach achieved comparable performance to supervised methods on dataset from four different domains, which suggests that taking advantages of both domain-independent and dependent knowledge may benefit general domain sentiment classification. Li et al. [2010] exploited a two-view model, i.e. personal and impersonal view, for sentiment classification. Specifically, personal views consist of sentences which directly express one's feeling and preference towards a target object; whereas impersonal views tend to provide objective domain-specific knowledge. For example, the sentence "*I love this book.*" is a personal view which contains domain-independent clue "*love*" that are highly relevant to sentiment classification. By contrast, an impersonal view "*It is too small.*" describes one's objective evaluation of a target object. It was reported that by employing the two views which were extracted with unsupervised mining, significant performance gain can be achieved with semi-supervised learning using the Co-Training algorithm. More recently, He et al. [2011] proposed to augment polarity-bearing topics for cross-domain sentiment classification. They first extracted polarity-bearing topics using the joint sentiment-topic (JST) model [Lin et al., 2011b] by incorporating a domain-independent sentiment lexicon; the original document spaces were then augmented with those extracted topics to form a new document representation as input to a Maximum Entropy classifier. They achieved state-of-the-art in-domain supervised classification accuracy in both the movie review and the multi-domain sentiment (MDS) datasets (more than 90% on average) and outperformed SCL [Blitzer et al., 2007] in cross-domain sentiment classification.

There has also been work on addressing domain adaptation by exploring careful structuring of features. Dasgupta and Ng [2009a] first mined the unambiguous reviews using spectral clustering, and then exploited those unambiguous reviews to classify the ambiguous reviews via a combination of active learning, transductive learning and ensemble learning. The more recently proposed spectral feature alignment (SFA) algorithm [Pan et al., 2010] addresses the domain transfer problem by aligning domain-specific words from different domains into unified clusters, using domain-independent words as a bridge. Leveraging labelled data only from the source domains, SFA outperformed most previous approaches in cross domain sentiment classification [Blitzer et al., 2007; Dasgupta and Ng, 2009a; He et al., 2011; Li et al., 2010].

2.1.2.2 Cross-lingual Sentiment Classification

In contrast to monolingual sentiment adaptation which addresses the domain mismatch issue for sentiment classification (e.g. Book vs. Electronics reviews), cross-lingual sentiment classification focuses on the mismatch arises from language differences, that is to use labelled data from a source language to build a sentiment classifier for a different target language. Semi-supervised techniques have also been widely applied to the task of cross-lingual sentiment classification, owning to the fact that some languages (typically English) have much richer sentiment resources (e.g., labelled corpus and lexicon) than others.

Most cross-lingual sentiment classification approaches focus on transferring sentiment or subjectivity analysis capabilities from English to the language that lacks labelled data for training. Mihalcea et al. [2007] explored two approaches for developing subjectivity classifiers in Romanian leveraging English resources. The lexicon-based approach used a subjectivity lexicon translated from an English one; whereas the corpus-based approach used an English subjectivity classifier and a manually translated parallel corpus for classifier training. It was found that the corpus-based approach outperformed the lexicon-based approach which was low in recall. Banea et al. [2008] also leveraged available English resources to generate resources for subjectivity classification in both Romanian and Spanish. Rather than relying on manual translation, they instead automatically translated the English resources by exploiting machine translation engines, and the classification performance was comparable to the approach using manually translated corpora [Mihalcea et al., 2007].

Leveraging English resources for Chinese sentiment classification has also been studied. Wan [2009] proposed to use the Co-Training algorithm for Chinese sentiment classification, which makes use of annotated English reviews and some amounts of unlabelled Chinese reviews as training data. Akin to Banea et al. [2008], the language gap was overcome with the use of readily available machine translation services. The proposed approach achieved more than 80% accuracy in sentiment classification, which outperformed the standard inductive and transductive SVM classifiers. Lu et al. [2011] proposed a maximum entropy-based EM algorithm for joint bilingual sentiment classification at the sentence level. In contrast to Wan [2009] focusing on improving sentiment classification performance in one language, the approach of Lu et al. [2011] can simultaneously learn better monolingual sentiment classifiers for both English and Chinese by exploiting an unlabelled parallel corpus together with labelled data in each language.

More recently, there is a study investigating the use of machine translation for cross-lingual sentiment classification [Duh et al., 2011]. It was argued that machine translation is ripe for cross-lingual sentiment classification, but the cross-lingual classification problem itself is qualitatively different from the monolingual sentiment adaptation problem which requires better understanding of its special characteristics.

2.1.3 Unsupervised and Weakly-supervised Approaches

The supervised and semi-supervised approaches discussed in Sections 2.1.1 and 2.1.2 can be categorized as corpus-based methods as they use a labelled or unlabelled data to train sentiment classifiers. Given the difficulties of supervised and semi-supervised sentiment analysis, it is conceivable that unsupervised or weakly-supervised approaches to sentiment classification are even more challenging. Nevertheless, solutions to unsupervised or weakly-supervised sentiment classification are of practical significance owing to its domain-independent nature [Dasgupta and Ng, 2009b].

In the absence of annotated data, unsupervised sentiment classification typically employs a very small number of sentiment seed words as reference features for polarity identification [Read and Carroll, 2009; Turney and Littman, 2002]. The pioneering work in this line is that of Turney and Littman [2002], in which a document is classified as positive or negative by the average sentiment orientation of the phrases containing adjectives or adverbs in the document. The sentiment orientation of a phrase is calculated as the pointwise mutual information (PMI) with a positive word "*excellent*" minus the PMI with a negative word "*poor*". The proposed approach achieved an accuracy of 84% for automobile reviews and 66% for movie reviews. In the same vein, Read and Carroll [2009] measured the similarity between words and polarity prototypes such as "*excellent*" and "*good*" with three different methods, namely, lexical association (using PMI), semantic spaces, and distributional similarity. While Turney and Littman [2002] only used one polarity prototype for each sentiment class, Read and Carroll experimented with seven polarity prototypes obtained from Roget's Thesaurus and WordNet[1] through a selection process based on their frequency in the Gigaword corpus. The best result was achieved using PMI with 69.1% accuracy obtained on the movie review data.

[1] `http://wordnet.princeton.edu/`

While a fixed number of sentiment seed words have been used in the aforementioned work [Read and Carroll, 2009; Turney and Littman, 2002], there have been attempts to incrementally enlarge the unlabelled examples with self-training based on the original seed word input [Zagibalov and Carroll, 2008a,b]. Starting with a single Chinese sentiment seed word meaning "*good*", Zagibalov and Carroll [2008b] used iterative retraining to gradually enlarge the seed vocabulary. Those enlarged sentiment-bearing words are selected based on their relative frequency in both the positive and negative parts of the current training data. The sentiment orientation of a document is then determined by the sum of the sentiment scores of all the sentiment-bearing lexical items found in the document. Problems with this approach are that there is no principled mechanism for determining the optimal iteration number for training as well as for selecting the initial seed word, where inappropriate seed word selection may result in very poor accuracy. As such, in subsequent work, Zagibalov and Carroll [2008a] introduced a way for automatic seed word selection based on some heuristic knowledge, and an iteration control method was proposed so that iterative training stops when there is no change to the classification of any document over the previous two iterations. However, this strategy does not necessarily permit the best classification accuracy. Instead of using sentiment seed words, Dasgupta and Ng [2009b] proposed an unsupervised sentiment classification algorithm where user feedback is provided in a spectral clustering process in an interactive manner to ensure that texts are clustered along the sentiment dimension. Features induced for each dimension of spectral clustering can be considered as sentiment-oriented topics. They achieved 70.9% and 69.3% classification accuracy on the movie review and multi-domain sentiment datasets respectively. Nevertheless, human identification of the most important dimensions during spectral clustering is required.

Weakly-supervised sentiment classification approaches are mostly lexicon-based, some of which integrate with corpus-based methods as a hybrid model [Andreevskaia and Bergler, 2008; Melville et al., 2009; Qiu et al., 2009; Tan et al., 2008]. Compared to the seed words used in unsupervised methods, the sentiment lexicon, consisting of a list of positive and negative sentiment bearing words, is usually much larger in size and is used as reference features for sentiment classification. Such sentiment lexicons can be constructed from domain-independent sources in many different ways, ranging from manual approaches [Whitelaw et al., 2005], to semi-automated approaches [Abbasi et al., 2008; Argamon et al., 2009; Kim and Hovy, 2004], and even almost fully automated approaches [Kaji and Kitsuregawa, 2006; Kanayama and Nasukawa, 2006; Turney and Littman, 2002]. Analogous to the unsupervised approach that uses iterative retraining [Zagibalov and Carroll, 2008b], Qiu et al. [2009] also used a lexicon-based iterative process to iteratively enlarge an initial sentiment dictionary from the first phrase. But instead of using a single seed word as Zagibalov and Carroll [2008b], they started with a much larger Chinese sentiment dictionary *HowNet*[1] as the initial lexicon. Documents classified from the first phase were taken as a training set to train SVMs, which were subsequently used to revise the results produced from the first phase. This self-supervised approach was tested on reviews from ten different domains, and outperformed the best results of the approach by Zagibalov and Carroll [2008a] on the same data over 6% in F-measure.

In a similar vein, Andreevskaia and Bergler [2008] integrated a lexicon-based system trained on WordNet and a corpus-based classifier trained on a small set of annotated in-domain data for sentence-level sentiment annotation across different domains. Melville et al. [2009] combined lexical information from a sentiment lexicon with labelled documents, where word-class probabilities in a Naive Bayes classifier learning were calculated as a weighted combination of word-class distributions estimated from the sentiment lexicon and the labelled documents respectively. It was observed in both studies [Andreevskaia and Bergler, 2008; Melville et al., 2009] that a framework integrating both lexical knowledge and corpus training examples performs better than using lexical knowledge or training data in isolation.

2.1.4 Sentiment Dynamics and Subjectivity Detection

Previous sections have discussed various approaches to sentiment classification ranging from supervised, semi-supervised to unsupervised learning. In this section, we review some related work in sentiment dynamics and subjectivity detection as these two lines of work are closely related to the dynamic joint sentiment-topic (dJST) model and the subjective LDA (subjLDA) model proposed in this thesis, as described in Chapters 4 and 5, respectively.

2.1.4.1 Sentiment Dynamics

Most sentiment classification tasks assume that input data is time-invariant and the sentiment of data does not exhibit dynamics. However, in reality, sentiment distributions of online content evolve over time and exhibit strong correlation with its published time

[1] `http://www.keenage.com/download/sentiment.rar`

stamp. This observation has thus motivated work on modelling sentiment dynamics in time-variant datasets.

There has not been much 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 and isotonic regression. Their proposed method has mainly been tested for sentence-level sentiment flow prediction within a document. Mei et al. [2007] employed a hidden Markov model (HMM) to tag every word in the collection with a topic and sentiment polarity, where the topic and sentiment of each word were detected by a topic-sentiment mixture model beforehand. The topic life-cycles and sentiment dynamics were then computed by counting the number of words labelled with the corresponding state over time.

In a recent study, Bollen et al. [2009; 2010] showed that public mood patterns from a sentiment analysis of Twitter posts do relate to the 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., 1971] by simply matching the terms extracted from each tweet to the set of POMS mood adjectives without considering the individual topic of each tweet. Similar phenomena have also been observed in the research by Lux [2008], who studied the causal relationship between investors' mood and subsequent stock price changes, based on weekly survey data on the short-term and medium-term sentiment of German investors. Using the vector autoregression (VAR) model, it was found that either sentiment is exogenous and drives stock returns, or returns and sentiment define a simultaneous system with mutual causation.

2.1.4.2 Subjectivity Detection

A primary task of sentiment analysis is subjectivity detection, which automatically identifies whether a given piece of text expresses opinions or states facts. While sentiment classification and subjectivity detection are closely related to each other, it has been reported that separating subjective and objective instances from text is more difficult than sentiment classification, and the improvement of subjectivity detection can benefit the latter as well, because not every part of a document is equally informative for inferring the document sentiment [Mihalcea et al., 2007].

Compared to sentiment classification, work on subjectivity detection is mostly at sentence level and is often viewed as a text classification problem where a classifier is trained from an annotated corpus. An early work by Yu and Hatzivassiloglou [2003] built a subjectivity classification system for opinion question answering. Using various n-gram features and a polarity lexicon, the proposed system was able to perform subjectivity classification at both document and sentence level. Pang and Lee [2004] separated subjective sentences from objective ones using a graph-based minimum cut algorithm, where subjective extractions were subsequently used as input for sentiment classifier training which yielded better sentiment accuracy than training on the whole document.

Riloff and Wiebe [2003] proposed a bootstrapping method for sentence-level subjectivity detection. They started with high-precision rule-based subjectivity classifiers which automatically identified subjective and objective sentences in un-annotated texts. The subjective expression patterns were learned from syntactic structure output from the previously labelled high confidence texts. The learned patterns were used to automatically identify additional subjective sentences which enlarged the training set, and the entire process was then iterated. In subsequent work, Wiebe and Riloff [2005] performed subjectivity detection using a method similar to Riloff and Wiebe [2003], but moved one step forward in that they also learned objective expressions apart from subjective expressions. As the subjective/objective expression patterns are based on syntactic structures, they are more flexible than unigrams or n-grams.

Wilton and Raaijmakers [2008] compared the performance of classifiers trained using word n-grams, character n-grams, and phoneme n-grams for recognizing subjective utterances in multiparty conversation. They found that using character n-grams from the reference transcriptions gave the best results, significantly outperforming word n-grams in terms of subjective recall and F-measure. Raaijmakers et. al [2008] extended the work of Wilson and Raaijmakers [2008] and further analysed the performance of detecting subjectivity in meeting speech by combining a variety of multimodal features including additional prosodic features. They found that the combination of all features performed best and that the prosodic features were less useful in discriminating between positive and negative utterances. More recently, Murray and Carenini [2009] proposed to learn subjective patterns from both labelled and unlabelled data using n-gram word sequences with varying levels of lexical instantiation. Their approach for learning subjective patterns relies on n-grams, which is similar to Raaijmakers et. al [2008], but goes beyond fixed sequences of words by allowing lexical instantiation in varying levels and thus offers more flexibility.

In contrast to most of the aforementioned methods for subjectivity detection relying on either labelled corpora or linguistic pattern extraction for subjectivity classifier training [Murray and Carenini, 2009; Pang and Lee, 2004; Riloff and Wiebe, 2003; Wiebe and Riloff, 2005; Wilson and Raaijmakers, 2008], the subjectivity detection LDA (subjLDA) model [Lin et al., 2011a] proposed in this thesis formulates the subjectivity detection problem as a weakly-supervised generative model learning. Apart from being able to achieve comparable performance to previous approaches using supervised learning [Wiebe and Riloff, 2005], subjLDA is relatively simple and uses only a small set of subjectivity clues as supervised information. Detailed discussion of the subjLDA model will be given in Chapter 5.

2.1.5 Discussion

Sentiment analysis has been an active research area in recently years and has seen many useful applications such as in business intelligence and social economics. Compared to the traditional text classification which focuses on topic identification, sentiment analysis is deemed to be more difficult which poses intellectual challenges in several aspects [Pang and Lee, 2008]:

- Sentiment is often embodied in subtle linguistic mechanisms such as the use of sarcasm, which makes sentiment classification difficult.
- Sentiment is domain-dependent, where sentiment expressions in different domains can be quite different. Besides, even for data from the same domain, sentiment distributions may vary over time, especially for collections that span years or decades [Read, 2005]. These factors have made the development of a sentiment system for general domains very challenging.
- In contrast to topic-based text categorization, word order in sentiment classification is of great importance, as the order in which different opinions are presented can result in a completely opposite overall sentiment polarity. So while generally satisfactory performance can be achieved using bag-of-words feature representation for topic categorization, it may not be sufficient for sentiment classification and modelling deeper linguistic knowledge is needed for improving classification accuracy.

Driven by these challenges, recent years have seen advances of different machine learning techniques in sentiment analysis.

Supervised methods can usually achieve good performance when training data is sufficiently large. However, there are several associated issues. First, supervised classifiers are typically domain-dependent, where classifiers trained on one domain lose accuracy when applied on test data whose distributions differ significantly from the training data. Second, online content varies widely across domains, and in-domain features also evolve over time, which makes the cost of annotating corpora for each domain unrealistic. As a result, there has been active research for developing semi-supervised or unsupervised learning methods for sentiment analysis.

Semi-supervised approaches fall between supervised and unsupervised approaches by leveraging a large amount of unlabelled data and a small amount of labelled data for training. The main advantages of semi-supervised learning are: (1) only a small amount of labelled data is needed, so the labelling cost is much less expensive than supervised approaches; (2) it has been found in many machine learning applications that unlabelled data used in conjunction with a small amount of labelled data can produce considerable improvement in learning accuracy [Bishop, 2006]. Therefore, there has been increasing interest in exploiting semi-supervised learning for domain adaptation and cross-lingual sentiment classification, where some of the approaches even achieve comparable performance to fully supervised learning.

Unsupervised or weakly-supervised sentiment analysis techniques are promising as they can avoid the domain-dependency problem typically associated with supervised approaches. As an unsupervised classifier will not be able to identify which features are relevant to polarity classification in the absence of annotated data, unsupervised approaches normally resort to sentiment lexicons as a form of prior polarity knowledge for model learning. Such domain-independent sentiment lexicons can be acquired automatically or semi-automatically with much less effort compared to labelling a large training dataset. Nevertheless, the performance of unsupervised or weakly-supervised sentiment approaches is still inferior to supervised or semi-supervised learning.

2.2 Generative Topic Models

In the field of information retrieval (IR), a variety of probabilistic models have been used to analyse the content of documents and the meaning of words [Blei et al., 2003; Hofmann, 1999]. These models all share the idea that documents are mixtures of topics, where a topic is a multinomial distribution over words. By assigning high probability to a set of

Table 2.1: An illustration of four topic examples extracted from *The American Political Science Review* by LDA.

Topic 1	Topic 2	Topic 3	Topic 4
vote	court	states	economic
election	committee	war	model
party	congress	international	political
presidential	house	military	data
candidates	members	soviet	spending
elections	courts	state	variables
candidate	supreme	foreign	results
voters	cases	crisis	time
campaign	congressional	force	change

topic words that tend to co-occur in the same document, one can easily interpret what that topic is about as these co-occurring words tend to have some tight semantic relations with one another. Such topic clusters extracted from document collections are useful in many tasks, such as identifying the contents of those documents, retrieving the most relevant documents given a query, tracking changes in contents over time and measuring the similarities between documents. Table 2.1 shows four topic examples derived from *The American Political Science Review*[1] using latent Dirichlet allocation (LDA) [Blei et al., 2003] with the topic number being set to 20. We show the top 15 words that have the highest probability under each topic. For instance, seeing the co-occurring words such as *vote*, *election*, *party* and *candidates* under topic 1, one can easily figure out that the topic is about political elections; likewise, topic words such as *war*, *international*, *military* and *soviet* appearing together under topic 3 present clear clues that the topic relates to war and international affairs.

One of the leading topic models is the so called probabilistic latent semantic indexing (pLSI) model (also known as the aspect model) introduced by Hofmann [1999] for document modelling. pLSI is based on a mixture decomposition derived from a latent class model. In pLSI, each word in a document is modelled as a sample from a mixture model where the mixture components are multinomial random variables that can be viewed as representations of topics. Thus, different words in a document can be generated from different topics, and a document can be represented as the mixing proportions of those mixture components. However, pLSI is still an incomplete generative model in that there is no probability process for defining how the weights of the mixture components for each document are generated, making it difficult to directly apply the learned models to new documents. In this regard, the latent Dirichlet allocation (LDA) model proposed by Blei

[1] `http://www.cs.princeton.edu/~blei/topicmodeling.html`

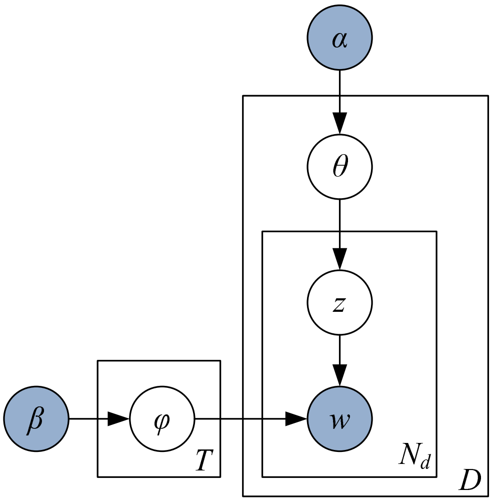


Figure 2.1: LDA model.

et al. [2003] solves the problem of pLSI by defining a generative model (i.e., a Dirichlet prior) over the mixture topic proportions for each document. Being able to use powerful statistical learning to infer latent semantic structures of text data, LDA has received considerable research interest and has been the foundation of developing numerous statistical models representing richer structures of language.

As LDA provides the basis of the three hierarchical Bayesian models proposed in the thesis, we give a detailed discussion of LDA in the following section.

2.2.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA), originally introduced by Blei et al. [2003], represents documents as random mixtures over latent topics, where each topic is a specialised distribution over words. A graphical model of LDA is shown in Figure 2.1, wherein nodes are random variables and edges indicate the dependence between nodes [Steyvers and Griffiths, 2007]. As a directed graph, arrows are used to indicate the direction of the relationship among nodes, so that arrows point to the *child* node from their direct *parent* node. Also with the standard graphical notation, the shaded and unshaded variables correspond to observed and hidden variables respectively, and the plates represent replicated sampling steps during the generative process [Bishop, 2006].

Given a corpus with a collection of D documents denoted by $C = \{d_1, d_2, ..., d_D\}$, each document in the corpus is a sequence of N_d words denoted by $d = (w_1, w_2, ..., w_{N_d})$, and each word in the document is an item from a vocabulary index with V distinct terms denoted by $\{1, 2, ..., V\}$. Also, let T be the total number of topics. The procedure for generating a word w_i in document d by LDA can be boiled down to two stages. First, one chooses a topic z_i from the per-document topic proportion $\boldsymbol{\theta}_d$. Following that, one draws a word w_i from the per-corpus topic-specific word distribution $\boldsymbol{\varphi}_{z_i}$. The formal definition of the generative process in LDA corresponding to the graphical model shown in Figure 2.1 is as follows:

- For each topic $j \in \{1, ..., T\}$
  - Draw $\boldsymbol{\varphi}_j \sim \text{Dir}(\boldsymbol{\beta})$.
- For each document $d \in \{1, ...D\}$
  - Draw a distribution $\boldsymbol{\theta}_d \sim \text{Dir}(\boldsymbol{\alpha})$.
  - For each word w_i in document d
    - Draw a topic $z_i \sim \text{Mult}(\boldsymbol{\theta}_d)$,
    - Draw a word $w_i \sim \text{Mult}(\boldsymbol{\varphi}_{z_i})$.

The goal of LDA is therefore to find a set of model parameters, here the per-document topic proportions $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_d\}_{d=1}^{D}$ and the per-corpus topic-word distributions $\boldsymbol{\Phi} = \{\boldsymbol{\varphi}_j\}_{j=1}^{T}$, that can best explain the observed words from the documents. However, exact inference in LDA is generally intractable, and we have to appeal to approximate inference algorithms for posterior estimation.

Approximate posterior inference algorithms generally fall into two categories: sampling approaches and optimization approaches [Hoffman et al., 2010]. Sampling approaches are mainly based on Markov Chain Monte Carlo (MCMC) methods, which can emulate high-dimensional probability distributions by the stationary distribution of a Markov chain. As a special case of MCMC, Gibbs sampling works in the way that the dimensions of a distribution are sampled sequentially one at a time, conditioned on the values of all other variables and data, and the posterior of interest can then be obtained from the Markov chain after it reaches the stationary state [Griffiths and Steyvers, 2004; Heinrich, 2005; Steyvers and Griffiths, 2007]. In contrast, optimization approaches are usually based on variational inference, which optimizes a simplified parametric distribution to be close in Kullback-Leibler divergence to the posterior [Jordan et al., 1999]. When used within a Bayesian hierarchical framework, variational inference is also called variational Bayes (VB). It has been shown in a previous study [Heinrich, 2005] that Gibbs sampling is relatively simple and yet efficient for high-dimensional models such as LDA. Thus, we choose Gibbs sampling as our model inference technique for all the three models we have developed in this thesis. For completeness, we give the full derivation of deriving a Gibbs sampler for LDA in the next section.

2.2.1.1 Model Inference

In LDA, our target inference is the posterior distribution $P(\mathbf{z}|\mathbf{w})$, i.e., the probability of corresponding topic assignments $\mathbf{z}$ given a corpus $\mathbf{w}$. As this distribution covers a large space of discrete random variables that are difficult to evaluate, a Gibbs sampler is used to simulate $P(\mathbf{z}|\mathbf{w})$ based on the full conditional distribution for a word token. Letting a word token with index $t = (d, n)$ denote the nth word position in the dth document, the full conditional distribution can be written as

$$P(z_t|\mathbf{z}^{\neg t}, \mathbf{w}) = \frac{P(\mathbf{w}, \mathbf{z})}{P(\mathbf{w}, \mathbf{z}^{\neg t})} = \frac{P(\mathbf{w}|\mathbf{z})}{P(\mathbf{w}^{\neg t}|\mathbf{z}^{\neg t})P(w_t)} \cdot \frac{P(\mathbf{z})}{P(\mathbf{z}^{\neg t})}, \tag{2.1}$$

where the superscript $\neg t$ denotes a quantity that excludes data with index t. The hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are omitted from Equation 2.1, but they provide a form of smoothing to the topic distribution. Specifically, the parameter components α_j and $\beta_{j,i}$ can be respectively interpreted as the prior observation counts for the number of times topic j is sampled from a document and the number of times word w_i is sampled from topic j, before having observed any actual words. Although $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ can be set as an asymmetric Dirichlet prior, a common practice is to use the symmetric setting where the parameter components have equal value, i.e., $\{\alpha_j\}_{j=1}^{T} = \alpha$ and $\{\{\beta_{ji}\}_{j=1}^{T}\}_{i=1}^{V} = \beta$. In the remainder of this section, we use a symmetric α and β for the LDA Gibbs sampler derivation.

To derive the full conditional distribution in Equation 2.1, we need to evaluate the joint distribution $P(\mathbf{w}, \mathbf{z})$, which can be factorized into two terms with $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ appearing in the first and second term, respectively:

$$\begin{aligned} P(\mathbf{w}, \mathbf{z}) &= P(\mathbf{w}|\mathbf{z})P(\mathbf{z}) \\ &= \int P(\mathbf{w}|\mathbf{z}, \boldsymbol{\Phi})P(\boldsymbol{\Phi}|\boldsymbol{\beta})\, d\boldsymbol{\Phi} \cdot \int P(\mathbf{z}|\boldsymbol{\Theta})P(\boldsymbol{\Theta}|\boldsymbol{\alpha})\, d\boldsymbol{\Theta}. \end{aligned} \tag{2.2}$$

For the first term in Equation 2.2, by integrating out $\mathbf{\Phi}$, we obtain

$$\begin{aligned}
P(\mathbf{w}|\mathbf{z}) &= \int P(\mathbf{w}|\mathbf{z}, \mathbf{\Phi}) P(\mathbf{\Phi}|\boldsymbol{\beta})\, d\mathbf{\Phi} && (2.3)\\
&= \int \prod_{j=1}^{T} \prod_{i=1}^{V} \varphi_{j,i}^{N_{j,i}} \frac{\Gamma(\sum_{i=1}^{V} \beta_{j,i})}{\prod_{i=1}^{V} \Gamma(\beta_{j,i})} \prod_{i=1}^{V} \varphi_{j,i}^{\beta_{j,i}-1}\, d\boldsymbol{\varphi}_j && (2.4)\\
&= \prod_{j=1}^{T} \frac{\Gamma(\sum_{i=1}^{V} \beta_{j,i})}{\prod_{i=1}^{V} \Gamma(\beta_{j,i})} \frac{\prod_{i=1}^{V} \Gamma(N_{j,i} + \beta_{j,i})}{\Gamma(\sum_{i=1}^{V} N_{j,i} + \beta_{j,i})} && (2.5)\\
&= \left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V}\right)^T \prod_j \frac{\prod_i \Gamma(N_{j,i} + \beta)}{\Gamma(N_j + V\beta)}, && (2.6)
\end{aligned}$$

where $N_{j,i}$ is the number of times word i is associated with topic j, N_j is the number of times words are assigned to topic j in the corpus, and Γ is the gamma function.

Similarly for the second term, by integrating out $\mathbf{\Theta}$, we obtain

$$\begin{aligned}
P(\mathbf{z}) &= \int P(\mathbf{z}|\mathbf{\Theta}) P(\mathbf{\Theta}|\boldsymbol{\alpha})\, d\mathbf{\Theta} && (2.7)\\
&= \int \prod_{d=1}^{D} \prod_{j=1}^{T} \theta_{d,j}^{N_{d,j}} \frac{\Gamma(\sum_{j=1}^{T} \alpha_j)}{\prod_{j=1}^{T} \Gamma(\alpha_j)} \prod_{j=1}^{T} \theta_{d,j}^{\alpha_j - 1}\, d\boldsymbol{\theta}_d && (2.8)\\
&= \prod_{d=1}^{D} \frac{\Gamma(\sum_{j=1}^{T} \alpha_j)}{\prod_{j=1}^{T} \Gamma(\alpha_j)} \frac{\prod_{j=1}^{T} \Gamma(N_{d,j} + \alpha_j)}{\Gamma(\sum_{j=1}^{T} N_{d,j} + \alpha_j)} && (2.9)\\
&= \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^D \prod_d \frac{\prod_j \Gamma(N_{d,j} + \alpha)}{\Gamma(N_d + T\alpha)}, && (2.10)
\end{aligned}$$

where D is the total number of documents in the collection, $N_{d,j}$ is the number of times a word from document d is associated with topic j, and N_d is the total number of words in document d.

Combining Equations 2.6 and 2.10 with Equation 2.1 yields the expression for the full conditional distribution[1] from which the Gibbs sampler draws the hidden variable z_t for the word token $w_t = i$:

$$\begin{aligned}
P(z_t = j|\mathbf{w}, \mathbf{z}^{\neg t}) &= \frac{P(\mathbf{w}, \mathbf{z})}{P(\mathbf{w}, \mathbf{z}^{\neg t})} = \frac{P(\mathbf{w}|\mathbf{z})}{P(\mathbf{w}^{\neg t}|\mathbf{z}^{\neg t}) P(w_t)} \cdot \frac{P(\mathbf{z})}{P(\mathbf{z}^{\neg t})}\\
&\propto \frac{\Gamma(N_{j,i} + \beta)\Gamma(N_j^{\neg t} + V\beta)}{\Gamma(N_{j,i}^{\neg t} + \beta)\Gamma(N_j + V\beta)} \cdot \frac{\Gamma(N_{d,j} + \alpha)\Gamma(N_d^{\neg t} + T\alpha)}{\Gamma(N_{d,j}^{\neg t} + \alpha)\Gamma(N_d + T\alpha)} && (2.11)\\
&\propto \frac{N_{j,i}^{\neg t} + \beta}{N_j^{\neg t} + V\beta} \cdot \frac{N_{d,j}^{\neg t} + \alpha}{N_d^{\neg t} + T\alpha}. && (2.12)
\end{aligned}$$

[1] Applying $\Gamma(x+1) = x\Gamma(x)$, all the terms in Equation 2.11 are cancelled out except those that contain token t.

Using Equation 2.12, the Gibbs sampling procedure can then be run to sequentially sample all hidden variables from their distributions conditioned on the values of all other variables and data, until a stationary state of the Markov chain has been reached.

Having obtained samples from the Markov chain $\mathcal{M} = \{\mathbf{w}, \mathbf{z}\}$, the final task is to estimate the LDA model parameter sets $\{\boldsymbol{\Theta}, \boldsymbol{\Phi}\}$. According to the definition of multinomial distributions with Dirichlet prior [Bishop, 2006], the probability of $\boldsymbol{\theta}_d$ (i.e., the dth component of $\boldsymbol{\Theta}$) and $\boldsymbol{\varphi}_j$ (i.e., the jth component of $\boldsymbol{\Phi}$) given observations and hyperparameters can be written as:

$$P(\boldsymbol{\theta}_d|\mathcal{M}, \boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\theta}_d|\mathcal{N}_d + \boldsymbol{\alpha}) \tag{2.13}$$

$$P(\boldsymbol{\varphi}_j|\mathcal{M}, \boldsymbol{\beta}) = \mathrm{Dir}(\boldsymbol{\varphi}_j|\mathcal{N}_j + \boldsymbol{\beta}), \tag{2.14}$$

where $\mathcal{N}_d$ is the vector of topic observation counts for document d and $\mathcal{N}_j$ is the vector of term observation counts for topic j. The value of the Dirichlet random variables can then be obtained by using the expectation of the Dirichlet distribution.

The approximated per-document topic proportion is given by

$$\begin{align} \theta_{d,j} &= \int \theta_{d,j}\, \mathrm{Dir}(\boldsymbol{\theta}_d|\mathcal{N}_d + \boldsymbol{\alpha})\, d\boldsymbol{\theta}_d \tag{2.15}\\ &= \int \theta_{d,j} \frac{\Gamma(\sum_{j=1}^{T} N_{d,j} + \alpha_{d,j})}{\prod_{j=1}^{T} \Gamma(N_{d,j} + \alpha_{d,j})} \prod_{j=1}^{T} \theta_{d,j}^{\alpha_{d,j}-1}\, d\boldsymbol{\theta}_d \tag{2.16}\\ &= \frac{N_{d,j} + \alpha_{d,j}}{N_d + \sum_{j=1}^{T} \alpha_{d,j}} \tag{2.17}\\ &= \frac{N_{d,j} + \alpha}{N_d + T\alpha}. \tag{2.18} \end{align}$$

Similarly, the approximated per-corpus topic-word distribution is

$$\begin{align} \varphi_{j,i} &= \int \varphi_{j,i}\, \mathrm{Dir}(\boldsymbol{\varphi}_j|\mathcal{N}_j + \boldsymbol{\beta})\, d\boldsymbol{\varphi}_j \tag{2.19}\\ &= \int \varphi_{j,i} \frac{\Gamma(\sum_{i=1}^{V} N_{j,i} + \beta_{j,i})}{\prod_{i=1}^{V} \Gamma(N_{j,i} + \beta_{j,i})} \prod_{i=1}^{V} \varphi_{j,i}^{\beta_{j,i}-1}\, d\boldsymbol{\varphi}_j \tag{2.20}\\ &= \frac{N_{j,i} + \beta_{j,i}}{N_j + \sum_{i=1}^{V} \beta_{j,i}} \tag{2.21}\\ &= \frac{N_{j,i} + \beta}{N_j + V\beta}. \tag{2.22} \end{align}$$

2.2.2 Static Topic Models

Research in statistical models of word co-occurrence has led to the development of a variety extensions to LDA. For instance, by carefully designing the graphical model structure, models can be used to capture specific word co-occurrence dependencies in the data, such as models of words and their authors [Rosen-Zvi et al., 2004], of words and research paper citations [Erosheva et al., 2004], of words and social network entities and attributes [McCallum et al., 2005; Wang et al., 2005], and of words and Markov dependencies [Griffiths et al., 2005]. As it is assumed that the datasets to which these models are applied have static word co-occurrence patterns, these models are considered as static topic models.

Rosen-Zvi et al. [2004] proposed the author-topic (AT) model which models the relationships between authors, documents and topics. They assumed that each author is associated with a multinomial distribution over topics and each topic is associated with a multinomial distribution over words. By accounting for the authorship information of documents in the generative process, the AT model is able to reveal which authors have the highest impact on a particular topic. In a similar application domain, Erosheva et al. [2004] used topic models to analyse the semantic content of a document as well as its citations of other publications, so called the mixed membership models. By assuming a fixed number of internal categories with each being characterized by multinomial distributions over words and references, the proposed model can identify internal categories of publications and provide soft classifications of papers into these categories.

Rather than focusing on paper and citation data, there have been attempts to analyse people's roles and salient groups of entities in a social network [McCallum et al., 2005; Wang et al., 2005]. McCallum et al. [2005] proposed the author-recipient-topic (ART) model for social network analysis, which learns topic distributions based on the direction-sensitive messages sent between entities. By discovering the prominent topics of each author-recipient pair and the topics each person is most likely to send and receive, the ART model is able to predict people's roles in a social network, such as *administrative assistant* vs. *manager*. Another similar work is the Group-Topic (GT) model [Wang et al., 2005], which simultaneously discovers groups among the entities and topics among the corresponding text. One key feature of the GT model is that it captures the attributes associated with the interactions between entities, and uses distinctions of these attributes to better assign group memberships. In contrast to the ART model that clusters entities with similar roles, the GT model clusters entities into groups based on their relations to other entities.

While topic models have been widely applied to texts with the bag-of-words assumption that words within a document are exchangeable, capturing the linguistic structures of languages such as syntax is of practical importance for applications like word sense disambiguation [Boyd-Graber and Blei, 2007]. Griffiths et al. [2005] proposed a generative model which models both short-range syntactic dependencies and long-range semantic dependencies between words by combining a hidden Markov model (HMM) with a topic model. With a sentence being factorized into function words handled by the HMM, and the content words handled by the topic model, their approach is able to find syntactic classes and semantic topics simultaneously. The more recently proposed syntactic topic models (STM) [Boyd-Graber and Blei, 2010] also considers the local context of a document. But in contrast to the approach of Griffiths et al. [2005] which generates words either from the syntactic or thematic context with a linear sequence model, STM is based on a non-parametric Bayesian method which compounds both thematic and syntactical constraints during the word generation process, so that each word of a sentence is generated by a distribution that combines document-specific topic weights and parse-tree-specific syntactic transition. In this sense, the STM model is able to capture important connections preserved in a syntactic parse.

As shown in the aforementioned work [Erosheva et al., 2004; McCallum et al., 2005; Rosen-Zvi et al., 2004; Wang et al., 2005], topic models constructed for purpose-specific applications often involve incorporating metadata for model learning, which in general can be categorized into two types depending on how the metadata are incorporated. One type is the so called *downstream* topic models, where both words and metadata are generated simultaneously conditioned on the topic assignment of the document [Mimno and McCallum, 2008]. Example models of this type include the mixed membership model [Erosheva et al., 2004] and the GT model [Wang et al., 2005]. The *upstream* topic models, by contrast, start the generative process with the observed metadata elements, and represent the topic distributions as a mixture of distributions conditioned on the metadata elements. Example models of this type are the AT model [Rosen-Zvi et al., 2004] and the ART model [McCallum et al., 2005].

For both downstream and upstream models, most of the models are customized for a special type of metadata, lacking the capability to accommodate data type beyond their original intention. This limitation has thus motivated work on developing a generalized framework for incorporating metadata into topic models [Andrzejewski et al., 2011; Blei and McAuliffe, 2010; Mimno and McCallum, 2008]. The supervised latent Dirichlet allocation (sLDA) model [Blei and McAuliffe, 2010] addresses the prediction problem of review ratings by inferring the most predictive latent topics of document labels. In sLDA, metadata (i.e. the responses) are generated by learning the parameters of a generalized linear model, with a link function and an exponential family dispersion function specified by the modeler. This essentially makes the sLDA model a more flexible framework than other downstream models [Erosheva et al., 2004; Wang et al., 2005] because it is capable of accommodating various types of metadata such as unconstrained real values and class labels. With the goal of incorporating arbitrary types of feature, Mimno and McCallum [2008] proposed the Dirichlet-multinomial regression (DMR) topic model which includes a log-linear prior on the document-topic distributions, where the prior is a function of the observed document features. The intrinsic difference between DMR and its complement model sLDA lies in that, while sLDA treats observed features as generated variables, DMR considers the observed features as a set of conditioned variables. Therefore, while incorporating complex features may result in increasingly intractable inference in sLDA, the inference in DMR can remain relatively simple by accounting for all the observed metadata in the document-specific Dirichlet parameters. More recently, Andrzejewski et al. [2011] proposed the Fold.all (First-Order Logic latent Dirichlet ALLocation) model for incorporating general domain knowledge into LDA. In Fold.all, a domain expert first specifies the domain knowledge as First-Order Logic (FOL) rules, after which the model automatically incorporates domain knowledge into the LDA inference to produce topics shaped by both data and the rules. However, building such a complicated generalized framework results in some tradeoffs: first the rule weights cannot be learned automatically and must be assigned manually, and second the inference for a logic-based model is less efficient than a custom inference procedure tailored to the specific model.

2.2.3 Joint Sentiment and Aspect Models

In Section 2.2.2, we discussed a series of topic model variants, most of which are custom-built for incorporating metadata that are specific to a target domain. This section is dedicated to work on models which jointly model topics and a special type of domain knowledge *sentiment*. The rationale is that compared to knowledge like authorship information, citation data and social network attributes, etc., sentiment has distinct properties such as domain- and time-dependence, and has been shown to be more difficult to classify than topic-based text classification tasks [Pang et al., 2002]. In addition, work on jointly modelling topics and sentiment is closely related to the three topic models we have proposed in the thesis, and some of this prior research inspired our work.

Capturing the interactions between topics and sentiments plays an important role in sentiment analysis as sentiment polarities are topic- and domain-dependent [Eguchi and Lavrenko, 2006]. Although work on jointly modelling sentiment and topic is still relatively sparse, some studies [Mei et al., 2007; Titov and McDonald, 2008a,b] have focused on a similar vision. The Topic-Sentiment Model (TSM) [Mei et al., 2007] models a mixture of topics and predicts sentiment for the entire document, which is closely related to the JST model [Lin and He, 2009; Lin et al., 2010, 2011b] proposed in the thesis. However, there are several intrinsic differences between JST and TSM. First, TSM is essentially based on the probabilistic latent semantic indexing (pLSI) [Hofmann, 1999] model with an extra background component and two additional sentiment components, whereas JST is based on LDA. Second, regarding the model generative process, TSM samples a word from the background component model if the word is a common English word. Otherwise, a word is sampled from either a topical model or one of the sentiment models (i.e., positive or negative sentiment model). Thus, in TSM the word generation for positive or negative sentiment is not conditioned on topic. This is a crucial difference compared to the JST model because in JST one draws a word from the distribution over words jointly conditioned on both topic and sentiment label. Third, for sentiment detection, TSM requires postprocessing to calculate the sentiment coverage of a document, while in JST the document sentiment can be directly obtained from the probability distribution of sentiment label given a document.

The Multi-Grain Latent Dirichlet Allocation model (MG-LDA) [Titov and McDonald, 2008a] and the Multi-Aspect Sentiment model (MAS) [Titov and McDonald, 2008b] are also closely related to JST. The MG-LDA model [Titov and McDonald, 2008a] is argued to be more appropriate to build topics that are representative of ratable aspects of customer reviews, by allowing terms to be generated from either a global topic or a local topic. Being aware of the limitation that MG-LDA is still purely topic based without considering the associations between topics and sentiments, Titov and McDonald further proposed the Multi-Aspect Sentiment model (MAS) [Titov and McDonald, 2008b] by extending the MG-LDA framework. The major improvement of MAS is that it can aggregate sentiment texts for the sentiment summary of each rating aspect extracted from MG-LDA. JST differs from MAS in several aspects. First, MAS works in a supervised setting requiring that every aspect is rated at least in some documents, which could be infeasible in real-world applications. By contrast, JST is weakly-supervised with only minimum prior information being incorporated, which in turn is more flexible. Second, the MAS model was designed for sentiment text extraction and aggregation, whereas JST is more suitable for the sentiment classification task.

More recently, there are several lines of work using topic models to discover aspects and aspect-specific opinion words from reviews [Brody and Elhadad, 2010; Zhao et al., 2010]. In order to extract aspect topics, both studies make the same assumption that each sentence is associated with a single aspect. Brody and Elhadad [2010] proposed a framework to extract aspects and detect aspect-specific opinion words in an unsupervised manner. They took a two-stage approach by first extracting local aspect words using the LDA model by treating each sentence as a document; afterwards, aspect-specific opinion words were identified by propagating the polarity scores of adjectives and building the conjunction graph. The MaxEnt-LDA hybrid model [Zhao et al., 2010] also models aspects and opinions; but instead of modelling aspect and sentiment in a cascaded procedure as Brody and Elhadad [2010], they modelled both simultaneously by incorporating a supervised maximum entropy model into an unsupervised topic model. By assuming that aspect words and opinion words play different syntactic roles in a sentence, they trained a MaxEnt component with lexical and POS features to distinguish between aspect and opinion words, which subsequently allows a word to be generated in different ways, i.e., from a background model, general/specific aspect models or general/specific opinion models.

One intrinsic difference between JST and the aforementioned models [Brody and Elhadad, 2010; Zhao et al., 2010] is that, while the aspect and aspect-specific opinion words detected by these two models are clustered in separate topics, in JST sentiment and topic words are discovered simultaneously to form a sentiment-bearing topic, which can be used to capture sentiment association among words from different domains to overcome the data distribution differences. Such sentiment-bearing topics extracted by JST have been used for cross domain sentiment classification with promising results being achieved [He et al., 2011].

The recently proposed Aspect and Sentiment Unification Model (ASUM) [Jo and Oh, 2011] is very similar to JST, which detects sentiment and topics simultaneously by modelling each document with a sentiment distribution and a set of sentiment-specific topic proportions. The main difference is that while JST allows the words of a document to be sampled from different word distributions, ASUM constrains the model such that the words from the same sentence must be sampled from the same word distribution.

In Sections 2.2.2 and 2.2.3, we have discussed a wide range of topic models for various applications. These models, however, do not consider the dependencies between a document and its timestamp, assuming that documents fitted to the model have static co-occurrence patterns. So when fitting large archives of documents collected over a considerably long period of time, these models will not be able to reveal the time-variant patterns and may result in discovering less salient topics [Wang and McCallum, 2006]. Therefore, in Section 2.2.4, we introduce work on dynamic topic models which address the problem of modelling topically non-static data.

2.2.4 Dynamic Topic Models

Static topic models are powerful tools for analysing large collections of documents with the implicit assumption that documents are exchangeable [Blei and Lafferty, 2006]. In other words, the ordering of documents does not matter. However, this assumption may not be realistic for many datasets of interest such as product reviews and scholarly journals, which are often collected over time and reflect evolving contents. Therefore, recently there has been increasing interest in developing dynamic topic models to explicitly model the dynamics of topics exhibited in document collections [Blei and Lafferty, 2006; Iwata et al., 2010; Nallapati et al., 2007; Wang et al., 2009; Wang and McCallum, 2006].

There are two different views when modelling topic dynamics. One is to consider topics as mutable and model the trajectories of individual topics over time [Blei and Lafferty, 2006; Wang et al., 2009]; the other represents topics as constant and uses the time information to capture the changes in topic occurrence [Wang and McCallum, 2006]. The dynamic topic model (DTM) [Blei and Lafferty, 2006] uses a state space model, i.e., Kalman filter, to capture alignment among topics across different time steps. In order to model topic dynamics, documents in DTM are divided into sequential groups and within each group the documents are assumed exchangeable. Topics of a group slice then evolve from the topics of the previous slice governed by the state space model. The continuous time dynamic topic model (cDTM) [Wang et al., 2009] replaces the discrete state space model of DTM with a continuous generalization using Brownian motion. The major advantage of cDTM over DTM is that it can model sequential time-series data with arbitrary granularity, relaxing the constraint of DTM that an appropriate discrete time resolution must be chosen. While DTM and cDTM employ a Markov assumption over time that the distributions of current epoch only depend on the distributions of the previous epoch, 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. As such, TOT is able to create topics with narrow or broad time distributions, conditioned on whether the word co-occurrence pattern is observed for a brief moment or a consistent long time span.

In contrast to the work which analyse topic evolution in a certain time-scale of resolution [Blei and Lafferty, 2006; Wang et al., 2009; Wang and McCallum, 2006], some recent work evaluates topic dynamics in multiple time-scale of resolution [Iwata et al., 2010; Nallapati et al., 2007]. Nallapati et al. [2007] proposed the multiscale topic tomography model (MTTM) which employs non-homogeneous Poisson processes to model generation of word counts. Compared to DTM, cDTM and TOT, a novel feature offered by MTTM is that it can analyse topic evolution at various time-scales of resolution using Haar wavelets, which allows users to examine topic evolution at a particular time resolution of interest. The online multiscale dynamic topic model (MDTM) [Iwata et al., 2010] also models topic evolution with multiple timescales. Rather than using the Poisson process as MTTM, a relatively simple Dirichlet-multinomial framework was exploited by MDTM which assumes that the current topic distributions over words are generated based on the multiscale word distributions of previous epochs. Experimental results show that MDTM outperforms MTTM, DMT and LDA in terms of predictive perplexity over four real world document collections.

The dynamic JST (dJST) model proposed in this thesis is partly inspired by the MTTM and MDTM models. Instead of detecting topic evolution alone, dJST moves a step forward that it can analyse both topic and sentiment dynamics by assuming that the current sentiment-topic word distributions are generated from the Dirichlet distributions parameterized by the word distributions of the documents from the past. As will be shown in Chapter 4, dJST outperforms the non-dynamic versions of JST in terms of both perplexity and sentiment classification on a real world dataset, which demonstrates the effectiveness of modelling dynamics.

Aside from extension of topic models, there has also been increasing interest in incorporating time dependencies into a 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.

2.2.5 Discussion

Probabilistic topic models have been widely used for managing, organizing and summarizing large archives of documents by discovering the hidden thematic structures, which are known as topics, from those documents. A simplest kind of topic model is latent Dirichlet allocation (LDA) [Blei et al., 2003], which assumes that each document is generated from a set of topics with different proportions and all the documents in the collection share the same set of topics. Although we have mainly investigated work on text-based applications using topic models, topic models can also be applied to non-text applications such as computer vision [Fei-Fei and Perona, 2005] and collaborative filtering [Marlin, 2004].

LDA was originally used to analyse the texts of documents. However, documents may contain additional information which we want to account for. Thus, numerous topic model variants have been developed based on LDA which take different types of metadata into account, such as the AT model [Rosen-Zvi et al., 2004] for authors and topic relationships, the mixed membership model [Erosheva et al., 2004] for papers and citations, the ART model [McCallum et al., 2005] for role discovery in social networks, and the MaxEnt-LDA model [Zhao et al., 2010] for product aspects and sentiments. In order to capture the dependencies between word co-occurrences and metadata, all these models share some similar properties: (1) model graphical structures are carefully designed to cater for the specific metadata and application; and (2) the metadata accompanying with the documents is typically incorporated either in a *downstream* or *upstream* mode, depending on whether the metadata is generated conditioned on latent variables or the observed metadata elements initiate the generative process.

While most of the specially constructed models [Erosheva et al., 2004; McCallum et al., 2005; Rosen-Zvi et al., 2004; Zhao et al., 2010] cannot accommodate metadata beyond their original intention, there have been research efforts on developing generalized frameworks for accommodating features of various types [Andrzejewski et al., 2011; Blei and McAuliffe, 2010; Mimno and McCallum, 2008]. However, the benefits of framework generality and flexibility usually come at a cost such as increasingly intractable inference for the posterior, inference being less efficient than a customized inference procedure [Blei and McAuliffe, 2010], or requiring human intervention to define rule weights when incorporating domain knowledge [Andrzejewski et al., 2011].

On the other hand, the aforementioned topic models can be broadly categorized as static topic models, as they make the assumption that the permutations of the ordering of documents are invariant to the model inference. In other words, documents are exchangeable. When this assumption may be reasonable for modelling documents collected within a relatively short time span, it may be unrealistic when analysing collections that span years or decades. Therefore, there has been an increasing interest in developing dynamic topic models to detect topics from a collection and track how the topics evolve over time [Blei and Lafferty, 2006; Iwata et al., 2010; Wang et al., 2009; Wang and McCallum, 2006].

Different models hold different viewpoints in modelling topic dynamics and the time information, i.e., some model topics as mutable and change over time [Blei and Lafferty, 2006; Iwata et al., 2010; Wang et al., 2009], whereas others consider topics as constant and capture the occurrences of the topics themselves [Wang and McCallum, 2006]. While all these dynamic topic models have been shown able to detect improved topics and interpretable topic trends, it was argued that modelling topic dynamics with a particular viewpoint (e.g. model topics as mutable) may not necessarily be better than the other (e.g. model the occurrences of constant topics) [Wang and McCallum, 2006]. So when confronted with a new corpus and a new task, one has to decide which types of modelling assumption are appropriate depending on the goal to be achieved.

Chapter 3

Weakly-Supervised Joint Sentiment-Topic Model

3.1 Introduction

In this chapter, we focus on document-level sentiment classification for general domains in conjunction with topic detection and topic sentiment analysis, based on the proposed weakly-supervised joint sentiment-topic (JST) model [Lin and He, 2009]. This model extends the state-of-the-art topic model latent Dirichlet allocation (LDA) [Blei et al., 2003] by constructing an additional sentiment layer, assuming that topics are generated dependent on sentiment distributions and words are generated conditioned on the sentiment-topic pairs.

The JST model is distinguished from other related sentiment-topic models [Mei et al., 2007; Titov and McDonald, 2008b] in that: (1) JST is weakly-supervised, where the only supervision comes from a domain independent sentiment lexicon; (2) JST can detect sentiment and topics simultaneously. We suggest that the weakly-supervised nature of the JST model makes it highly portable to other domains for the sentiment classification task. While JST is a reasonable design choice for joint sentiment-topic detection, one may argue that the reverse is also true, namely that sentiments may vary according to topics. Thus, we also studied a reparameterized version of JST, called the Reverse-JST model, in which sentiments are generated dependent on topic distributions in the modelling process. It is worth noting that without a hierarchical prior, JST and Reverse-JST are essentially two reparameterizations of the same model.

Extensive experiments were conducted with both the JST and Reverse-JST models on the movie review (MR)[1] and multi-domain sentiment (MDS) datasets[2]. Although JST is equivalent to Reverse-JST without hierarchical priors, experimental results show that when sentiment prior information is encoded, these two models exhibit very different behaviours, with JST consistently outperforming Reverse-JST in sentiment classification. The portability of JST in sentiment classification is also verified by the experimental results on the datasets from five different domains, where the JST model even outperforms existing semi-supervised approaches in some of the datasets despite using no labelled documents. Aside from automatically detecting sentiment from text, JST can also extract meaningful topics with sentiment associations as illustrated by some topic examples extracted from the MR and MDS datasets.

We first present the overall structure of the JST model in Section 3.2. Then a reparameterized version of JST called the Reverse-JST model, obtained by reversing the sequence of sentiment and topic generation in the modelling process is studied in Section 3.3. Finally, Section 3.4 presents the experimental results of JST and Reverse-JST on document-level sentiment classification as well as topic extraction based on the movie review (MR) and the multi-domain sentiment (MDS) datasets.

3.2 Joint Sentiment-Topic (JST) Model

The existing framework of LDA has three hierarchical layers, where topics are associated with documents, and words are associated with topics. In order to model document sentiments, we propose a joint sentiment-topic (JST) model [Lin and He, 2009; Lin et al., 2011b] by adding an additional sentiment layer between the document and the topic layers. Hence, JST is effectively a four-layer model, where sentiment labels are associated with documents, under which topics are associated with sentiment labels and words are associated with both sentiment labels and topics. A graphical model of JST is represented in Figure 3.1.

Assume that we have a corpus with a collection of D documents denoted by $C = \{d_1, d_2, ..., d_D\}$; each document in the corpus is a sequence of N_d words denoted by $d = (w_1, w_2, ..., w_{N_d})$, and each word in the document is an item from a vocabulary index with V distinct terms denoted by $\{1, 2, ..., V\}$. Also, let S be the number of distinct sentiment labels, and T be the total number of topics. The formal definition of the generative process in JST corresponding to the graphical model shown in Figure 3.1 is as follows. Here $\boldsymbol{\lambda}$ is

[1] `http://www.cs.cornell.edu/people/pabo/movie-review-data`

[2] `http://www.cs.jhu.edu/~mdredze/datasets/sentiment/index2.html`

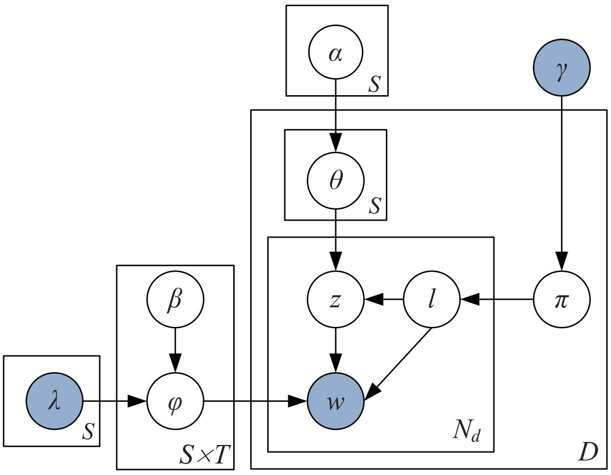


Figure 3.1: JST model.

the transformation matrix for encoding the prior knowledge from the sentiment lexicons; we will give a detailed discussion of $\boldsymbol{\lambda}$ in Section 3.2.1.

- For each sentiment label $l \in \{1, ..., S\}$
  - For each topic $j \in \{1, ..., T\}$,
    - draw $\boldsymbol{\varphi}_{lj} \sim \text{Dir}(\boldsymbol{\lambda}_l \cdot \boldsymbol{\beta}_{lj}^T)$
- For each document $d \in \{1, ..., D\}$,
  - choose a distribution $\boldsymbol{\pi}_d \sim \text{Dir}(\boldsymbol{\gamma})$
  - For each sentiment label l under document d
    - Choose a distribution $\boldsymbol{\theta}_{d,l} \sim \text{Dir}(\boldsymbol{\alpha})$
  - For each word w_i in document d
    - Choose a sentiment label $l_i \sim \text{Mult}(\boldsymbol{\pi}_d)$
    - Choose a topic $z_i \sim \text{Mult}(\boldsymbol{\theta}_{d,l_i})$
    - Choose a word $w_i \sim \text{Mult}(\boldsymbol{\varphi}_{l_i,z_i})$

The procedure for generating a word w_i in JST boils down to three stages. First, one chooses a sentiment label l from the per-document sentiment proportion $\boldsymbol{\pi}_d$. Following that, one chooses a topic z from the topic proportion $\boldsymbol{\theta}_{d,l}$, where $\boldsymbol{\theta}_{d,l}$ is conditioned on the sampled sentiment label l. It is worth noting that the topic proportion of JST is different from that of LDA. In LDA, there is only one topic proportion $\boldsymbol{\theta}_d$ for each document d. In contrast, in JST each document is associated with S (the number of sentiment labels)

Table 3.1: Parameter notations used in the JST model.

Symbol	*Description*
D	number of documents in the collection.
N_d	number of words in document d.
V	number of unique words in the corpus.
S	number of sentiment labels.
T	number of topics.
$\boldsymbol{\alpha}$	asymmetric Dirichlet priors on the mixing topic proportions, $\boldsymbol{\alpha} = \{\{\alpha_{l,z}\}_{z=1}^{T}\}_{l=1}^{S}$ ($S \times T$ matrix).
$\boldsymbol{\beta}$	asymmetric Dirichlet priors on the sentiment label and topic specific word distribution, $\boldsymbol{\beta} = \{\{\{\beta_{l,z,i}\}_{z=1}^{T}\}_{l=1}^{S}\}_{i=1}^{V}$ ($S \times T \times V$ matrix).
γ	symmetric Dirichlet priors on the mixing sentiment proportions (scalar).
$\boldsymbol{\pi}_d$	parameter notation for the sentiment mixing proportions for document d ($S-$ vector). For D documents, $\boldsymbol{\Pi} = \{\{\pi_{d,l}\}_{l=1}^{S}\}_{d=1}^{D}$ ($D \times S$ matrix).
$\boldsymbol{\theta}_{d,l}$	parameter notation for the topic mixing proportions for document d and sentiment label l ($T-$ vector). For D documents and S sentiment labels, $\boldsymbol{\Theta} = \{\{\{\theta_{d,l,z}\}_{z=1}^{T}\}_{l=1}^{S}\}_{d=1}^{D}$ ($D \times S \times T$ matrix).
$\boldsymbol{\varphi}_{l,z}$	parameter notation for the multinomial distribution over words for sentiment label l and topic z ($V-$ vector). For S sentiment labels and T topics, $\boldsymbol{\Phi} = \{\{\{\varphi_{l,z,i}\}_{l=1}^{S}\}_{z=1}^{T}\}_{i=1}^{V}$ ($S \times T \times V$ matrix)
$\boldsymbol{\lambda}$	parameter notation for the transformation matrix for encoding prior information ($S \times V$ matrix).

topic proportions, each of which corresponds to a sentiment label l with the same topics number T, i.e., $\boldsymbol{\Theta} = \{\{\boldsymbol{\theta}_{d,l}\}_{l=1}^{S}\}_{d=1}^{D}$. This feature essentially provides the means for the JST model to predict the sentiment associated with the extracted topics. Finally, one draws a word from the per-corpus word distribution $\boldsymbol{\varphi}_{z,l}$ which is conditioned on both topic z and sentiment label l. This is again different from LDA in that in LDA a word is sampled from the word distribution conditioned only on topic. The parameter notations used for the JST model are summarized in Table 3.1.

In our implementation, we used asymmetric priors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ and symmetric prior γ[1]. The hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ in JST can be respectively treated as the prior observation counts for the number of times topic z associated with sentiment label l is sampled from a document and the number of times words sampled from topic z are associated with sentiment label l, before having observed any actual words. Similarly, the hyperparameter γ can be interpreted as the prior observation counts for the number of times sentiment label l is sampled from a document before any word from the corpus is observed.

In JST, three sets of model parameters need to be inferred given the observed data,

[1] We have experimented symmetric and asymmetric settings with each of the hyperparameters. Empirical results show that asymmetric $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ with symmetric γ yields the best result.

namely, the per-document sentiment proportion $\boldsymbol{\pi}_d$, the per-document sentiment label specific topic proportion $\boldsymbol{\theta}_{d,l}$, and the per-corpus joint sentiment-topic word distribution $\boldsymbol{\varphi}_{l,z}$. Particularly, we will see later in this chapter that the per-document sentiment distribution $\boldsymbol{\pi}_d$ plays an important role in determining the document sentiment polarity.

3.2.1 Incorporating Model Priors

In contrast to the traditional topic-based classification, a fully unsupervised sentiment model will not be able to identify which features are relevant for polarity classification in the absence of annotated data [Pang and Lee, 2008]. Therefore, we introduced a mechanism for incorporating a small set of domain-independent sentiment-bearing words as prior knowledge (e.g., "*excellent*", "*awful*") for the JST and Reverse-JST model learning. Specifically, we constructed a dependency link of $\boldsymbol{\varphi}_{l,z}$ on the transformation matrix $\boldsymbol{\lambda} = \{\{\lambda_{l,i}\}_{i=1}^{V}\}_{l=1}^{S}$. The matrix $\boldsymbol{\lambda}$ modifies the Dirichlet priors $\boldsymbol{\beta}$, so that the word prior sentiment polarity can be captured.

The complete procedure for incorporating prior knowledge into the JST model is as follows. First, $\boldsymbol{\lambda}$ is initialized with all the elements taking a value of 1. Then for each term $w \in \{1, ..., V\}$ in the corpus vocabulary and for each sentiment label $l \in \{1, ..., S\}$, if w is found in the sentiment lexicon, the element $\lambda_{l,w}$ is updated as follows

$$\lambda_{lw} = \begin{cases} 0.9 & \text{if } S(w) = l \\ 0.05 & \text{otherwise} \end{cases}, \tag{3.1}$$

where the function $S(w)$ returns the prior sentiment label of w in a sentiment lexicon, i.e., neutral, positive or negative. For example, the word "*excellent*" with index i 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 $\lambda_{l,i}$ with $\beta_{l,z,i}$ (i.e. an element-wise multiplication), we can ensure that the word "*excellent*" has much higher probability of being drawn from the positive topic word distributions generated from a Dirichlet distribution with parameter $\boldsymbol{\beta}_{l_{pos}}$.

The previously proposed DiscLDA [Lacoste-Julien et al., 2008] and Labeled LDA [Ramage et al., 2009] models also utilize a transformation matrix to modify Dirichlet priors by assuming the availability of document class labels. DiscLDA uses a class-dependent linear transformation to project a K-dimensional (K latent topics) document-topic distribution into a L-dimensional space (L document labels), while Labeled LDA simply defines a one-to-one correspondence between LDA's latent topics and document labels. In contrast, we use word prior sentiment as supervised information and modify the topic-word Dirichlet priors for sentiment classification.

3.2.2 Model Inference

In order to estimate the parameter set $\{\mathbf{\Pi}, \mathbf{\Theta}, \mathbf{\Phi}\}$ in JST, we need to evaluate the posterior distribution $P(\mathbf{z}, \mathbf{l}|\mathbf{w})$, i.e., the probability of the corresponding topic assignments $\boldsymbol{z}$ and sentiment label assignments $\mathbf{l}$ given a corpus $\mathbf{w}$. As this distribution is difficult to evaluate directly due to large space of random variables, a Gibbs sampler is used to simulate the posterior distribution by the stationary distribution of a Markov chain.

For the nth word token in the dth document with an index $t = (d, n)$, the Gibbs sampler draws the hidden variables of interest, here z_t and l_t, from the full conditional distribution $P(l_t = k, z_t = j|\mathbf{w}, \mathbf{z}^{\neg t}, \mathbf{l}^{\neg t}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)$. Analogous to the derivation for a Gibbs sampler for LDA described in Section 2.2.1, the full conditional distribution can be derived from the joint distribution $P(\mathbf{w}, \mathbf{z}, \mathbf{l})$. Omitting the hyperparameters and using subscript $\neg t$ to indicate omitting the information of the word token with index t from the corresponding document, topic and sentiment label yields:

$$
\begin{aligned}
P(l_t = j, z_t = k|\mathbf{w}, \mathbf{z}^{\neg t}, \mathbf{l}^{\neg t}) &= \frac{P(\mathbf{w}, \mathbf{z}, \mathbf{l})}{P(\mathbf{w}, \mathbf{z}^{\neg t}, \mathbf{l}^{\neg t})} && (3.2)\\
&= \frac{P(\mathbf{w}|\mathbf{z}, \mathbf{l})P(\mathbf{z}, \mathbf{l})}{P(\mathbf{w}^{\neg t}|\mathbf{z}^{\neg t}, \mathbf{l}^{\neg t})P(w_t)P(\mathbf{z}^{\neg t}, \mathbf{l}^{\neg t})} && (3.3)\\
&\propto \frac{P(\mathbf{w}|\mathbf{z}, \mathbf{l})P(\mathbf{z}, \mathbf{l})}{P(\mathbf{w}^{\neg t}|\mathbf{z}^{\neg t}, \mathbf{l}^{\neg t})P(\mathbf{z}^{\neg t}, \mathbf{l}^{\neg t})} && (3.4)\\
&\propto \frac{P(\mathbf{w}|\mathbf{z}, \mathbf{l})}{P(\mathbf{w}^{\neg t}|\mathbf{z}^{\neg t}, \mathbf{l}^{\neg t})} \cdot \frac{P(\mathbf{z}|\mathbf{l})}{P(\mathbf{z}^{\neg t}|\mathbf{l}^{\neg t})} \cdot \frac{P(\mathbf{l})}{P(\mathbf{l}^{\neg t})}. && (3.5)
\end{aligned}
$$

The joint probability of the words, topics and sentiment label assignments can be factored into the following three terms:

$$
\begin{aligned}
P(\mathbf{w}, \mathbf{z}, \mathbf{l}) = P(\mathbf{w}|\mathbf{z}, \mathbf{l})P(\mathbf{z}, \mathbf{l}) &= P(\mathbf{w}|\mathbf{z}, \mathbf{l})P(\mathbf{z}|\mathbf{l})P(\mathbf{l}) && (3.6)\\
= \int P(\mathbf{w}|\mathbf{z}, \mathbf{l}, \mathbf{\Phi})P(\mathbf{\Phi}|\boldsymbol{\beta})\, d\mathbf{\Phi} &\cdot \int P(\mathbf{z}|\mathbf{l}, \mathbf{\Theta})\, P(\mathbf{\Theta}|\boldsymbol{\alpha})\, d\mathbf{\Theta} \cdot \\
\int P(\mathbf{l}|\mathbf{\Pi})\, P(\mathbf{\Pi}|\gamma)\, d\mathbf{\Pi}&, && (3.7)
\end{aligned}
$$

where each term contains only one model parameter and therefore can be handled separately.

For the first term, by integrating out $\mathbf{\Phi}$ using Dirichlet integrals, we obtain:

$$
\begin{aligned}
P(\mathbf{w}|\mathbf{z},\mathbf{l}) &= \int P(\mathbf{w}|\mathbf{z},\mathbf{l},\mathbf{\Phi})P(\mathbf{\Phi}|\boldsymbol{\beta})\,d\mathbf{\Phi} && (3.8)\\
&= \int \prod_{k=1}^{S}\prod_{j=1}^{T}\prod_{i=1}^{V} \varphi_{k,j,i}^{N_{k,j,i}} \frac{\Gamma(\sum_{i=1}^{V}\beta_{k,j,i})}{\prod_{i=1}^{V}\Gamma(\beta_{k,j,i})} \prod_{i=1}^{V} \varphi_{k,j,i}^{\beta_{k,j,i}-1}\, d\boldsymbol{\varphi}_{k,j} && (3.9)\\
&= \prod_{k=1}^{S}\prod_{j=1}^{T} \frac{\Gamma(\sum_{i=1}^{V}\beta_{k,j,i})}{\prod_{i=1}^{V}\Gamma(\beta_{k,j,i})}\cdot\\
&\quad \int \frac{\prod_{i=1}^{V}\Gamma(N_{k,j,i}+\beta_{k,j,i})}{\Gamma(\sum_{i=1}^{V}N_{k,j,i}+\beta_{k,j,i})} \frac{\Gamma(\sum_{i=1}^{V}N_{k,j,i}+\beta_{k,j,i})}{\prod_{i=1}^{V}\Gamma(N_{k,j,i}+\beta_{k,j,i})} \prod_{i=1}^{V} \varphi_{k,j,i}^{N_{k,j,i}+\beta_{k,j,i}-1}\, d\boldsymbol{\varphi}_{k,j}\\
& && (3.10)\\
&= \prod_{k=1}^{S}\prod_{j=1}^{T} \frac{\Gamma(\sum_{i=1}^{V}\beta_{k,j,i})}{\prod_{i=1}^{V}\Gamma(\beta_{k,j,i})} \frac{\prod_{i=1}^{V}\Gamma(N_{k,j,i}+\beta_{k,j,i})}{\Gamma(N_{k,j}+\sum_{i=1}^{V}\beta_{k,j,i})}, && (3.11)
\end{aligned}
$$

where $N_{k,j,i}$ is the number of times word i appeared in topic j with sentiment label k, $N_{k,j}$ is the number of times words are assigned to topic j and sentiment label k, and Γ is the gamma function.

For the second term, by integrating out $\mathbf{\Theta}$, we obtain:

$$
\begin{aligned}
P(\mathbf{z}|\mathbf{l}) &= \int P(\mathbf{z}|\mathbf{l},\mathbf{\Theta})\,P(\mathbf{\Theta}|\boldsymbol{\alpha})\,d\mathbf{\Theta} && (3.12)\\
&= \int \prod_{d=1}^{D}\prod_{k=1}^{S}\prod_{j=1}^{T} \theta_{d,k,j}^{N_{d,k,j}} \frac{\Gamma(\sum_{j=1}^{T}\alpha_{k,j})}{\prod_{j=1}^{T}\Gamma(\alpha_{k,j})} \prod_{j=1}^{T} \theta_{d,k,j}^{\alpha_{k,j}-1}\, d\boldsymbol{\theta}_{d,k} && (3.13)\\
&= \prod_{d=1}^{D}\prod_{k=1}^{S} \frac{\Gamma(\sum_{j=1}^{T}\alpha_{k,j})}{\prod_{j=1}^{T}\Gamma(\alpha_{k,j})} \frac{\prod_{j=1}^{T}\Gamma(N_{d,k,j}+\alpha_{k,j})}{\Gamma(N_{d,k}+\sum_{j=1}^{T}\alpha_{k,j})}, && (3.14)
\end{aligned}
$$

where D is the total number of documents in the collection, $N_{d,k,j}$ is the number of times a word from document d is associated with topic j and sentiment label k, and $N_{d,k}$ is the number of times sentiment label k is assigned to some word tokens in document d.

For the third term, integrating out $\mathbf{\Pi}$ yields:

$$
\begin{aligned}
P(\mathbf{l}) &= \int P(\mathbf{l}|\mathbf{\Pi})\,P(\mathbf{\Pi}|\gamma)\,d\mathbf{\Pi} && (3.15)\\
&= \int \prod_{d=1}^{D}\prod_{k=1}^{S} \pi_{d,k}^{N_{d,k}} \frac{\Gamma(\sum_{k=1}^{S}\gamma_k)}{\prod_{k=1}^{S}\Gamma(\gamma_k)} \prod_{k=1}^{S} \pi_{d,k}^{\gamma_k-1}\, d\boldsymbol{\pi}_d && (3.16)\\
&= \prod_{d=1}^{D} \frac{\Gamma(S\gamma)}{\Gamma(\gamma)^S} \frac{\prod_{k=1}^{S}\Gamma(N_{d,k}+\gamma)}{\Gamma(N_d+S\gamma)}, && (3.17)
\end{aligned}
$$

where N_d is the total number of words in document d.

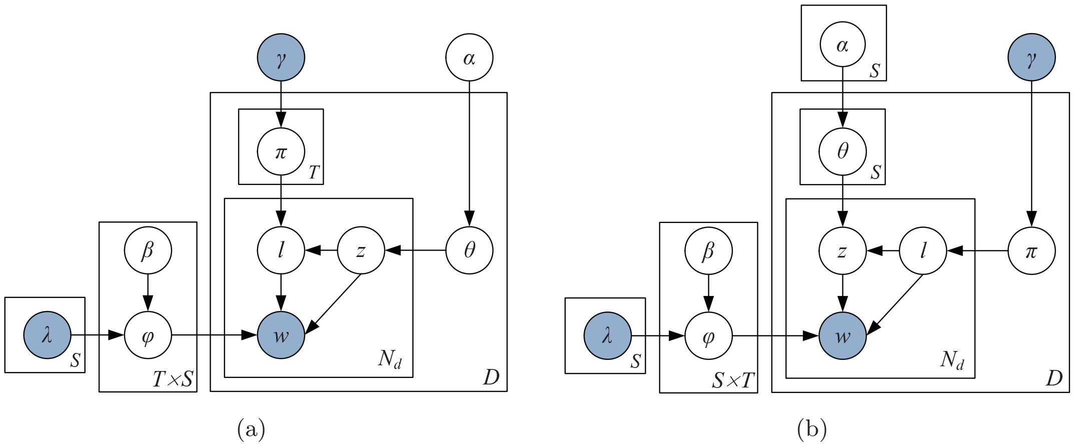


Figure 3.2: (a) Reverse-JST model; (b) JST model.

Substituting Equations 3.11, 3.14 and 3.17 into Equation 3.5 and cancelling out the terms that do not contain word token w_t yields the expression for the full conditional distribution, from which the Gibbs sampler draws the hidden variables z_t and l_t for the word token $w_t = i$:

$$P(z_t = j, l_t = k | \mathbf{w}, \mathbf{z}^{\neg t}, \mathbf{l}^{\neg t}) \propto \frac{N^{\neg t}_{k,j,i} + \beta_{k,j,i}}{N^{\neg t}_{k,j} + \sum_{i=1}^{V} \beta_{k,j,i}} \cdot \frac{N^{\neg t}_{d,k,j} + \alpha_{k,j}}{N^{\neg t}_{d,k} + \sum_j \alpha_{k,j}} \cdot \frac{N^{\neg t}_{d,k} + \gamma}{N^{\neg t}_{d} + S\gamma}. \tag{3.18}$$

Using Equation 3.18, the Gibbs sampling procedure can be run until a stationary state of the Markov chain has been reached. Samples obtained from the Markov chain are then used to estimate the JST model parameters according to the expectation of Dirichlet distribution (for detailed derivation please refer to Section 2.2.1):

The approximate per-corpus sentiment-topic word distribution is

$$\varphi_{k,j,i} = \frac{N_{k,j,i} + \beta_{k,j,i}}{N_{k,j} + \sum_{i=1}^{V} \beta_{k,j,i}}. \tag{3.19}$$

The approximate per-document sentiment label specific topic distribution is

$$\theta_{d,k,j} = \frac{N_{d,k,j} + \alpha_{k,j}}{N_{d,k} + \sum_{j=1}^{T} \alpha_{k,j}}. \tag{3.20}$$

Finally, the approximate per-document sentiment distribution is

$$\pi_{d,k} = \frac{N_{d,k} + \gamma}{N_d + S\gamma}. \tag{3.21}$$

The pseudocode for the Gibbs sampling procedure of JST is shown in Algorithm 1.

Algorithm 1 Gibbs sampling procedure of JST.

```
Input: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\gamma$, Corpus
Output: sentiment and topic label assignment for all word tokens in the corpus
Initialize sentiment and topic label assignment for each word token in the corpus with random sampling;
for $i = 1$ to max Gibbs sampling iterations do
  for all documents $d \in \{1, ..., D\}$ do
    for each word $t \in \{1, ..., N_d\}$ do
      Exclude word $t$ associated with sentiment label $l$ and topic label $z$ from counts $N_{k,j,i}$, $N_{k,j}$, $N_{d,k,j}$, $N_{d,k}$ and $N_d$;
      Sample a new sentiment-topic pair $\tilde{l}$ and $\tilde{z}$ using Equation 3.18;
      Update counts $N_{k,j,i}$, $N_{k,j}$, $N_{d,k,j}$, $N_{d,k}$ and $N_d$ using the new sentiment label $\tilde{l}$ and topic label $\tilde{z}$;
    end for
  end for
  for every 25 iterations do
    Update hyperparameter $\boldsymbol{\alpha}$ with maximum-likelihood estimation A;
  end for
  for every 100 iterations do
    Update matrices $\boldsymbol{\Phi}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\Pi}$ with new sampling results;
  end for
end for
```

3.3 Reverse Joint Sentiment-Topic (Reverse-JST) Model

In this section, we study a reparameterized version of the JST model called Reverse-JST, for which the underlying motivation is to investigate how the order of generating a sentiment label and a topic will affect the model performance in sentiment classification.

According to the graphic model shown in Figure 3.2b, the topic generation in JST is conditioned on sentiment labels, and then words are generated conditioned on the sampled sentiment-topic pairs. By contrast, in the Reverse-JST model as shown in Figure 3.2a, the sequence of generating a sentiment label and a topic is reversed, where sentiment labels are generated conditioned on topics, and then words are generated conditioned on both sentiment labels and topics. Using similar notations and terminologies to the JST model in Section 3.2, the joint probability of the words, the topics and sentiment label assignments of Reverse-JST can be factored into the following three terms:

$$P(\mathbf{w}, \mathbf{l}, \mathbf{z}) = P(\mathbf{w}|\mathbf{l}, \mathbf{z})P(\mathbf{l}, \mathbf{z}) = P(\mathbf{w}|\mathbf{l}, \mathbf{z})P(\mathbf{l}|\mathbf{z})P(\mathbf{z}). \tag{3.22}$$

As the Gibbs sampler for Reverse-JST can be derived in a similar way to JST, here we only give the full conditional distribution from which the Reverse-JST Gibbs sampler draws

Table 3.2: Dataset statistics. Note: †denotes before preprocessing and * denotes after preprocessing.

Dataset	# of words					
	MR	subjMR	MDS			
			Book	DVD	Electronics	Kitchen
Average doc. length†	666	406	176	170	110	93
Average doc. length*	313	167	116	113	75	63
Vocabulary size†	38,906	34,559	22,028	21,424	10,669	9,525
Vocabulary size*	25,166	18,013	19,428	20,409	9,893	8,512

the hidden variables z_t and l_t for a word token $w_t = i$:

$$P(z_t = j, l_t = k|\mathbf{w}, \mathbf{z}^{\neg t}, \mathbf{l}^{\neg t}) \propto \frac{N^{\neg t}_{j,k,i} + \beta_{j,k,i}}{N^{\neg t}_{j,k} + \sum_{i=1}^{V} \beta_{j,k,i}} \cdot \frac{N^{\neg t}_{d,j,k} + \gamma}{N^{\neg t}_{d,j} + S\gamma} \cdot \frac{N^{\neg t}_{d,j} + \alpha_j}{N^{\neg t}_{d} + \sum_{j=1}^{T} \alpha_j}. \tag{3.23}$$

As we do not have a direct per-document sentiment distribution in Reverse-JST, the distribution over sentiment labels for document $P(\mathbf{l}|d)$ is calculated based on the topic specific sentiment proportion $\boldsymbol{\pi}_{d,z}$ and the per-document topic proportion $\boldsymbol{\theta}_d$ as follows:

$$P(\mathbf{l}|d) = \sum_{z} P(\mathbf{l}|z, d)P(z|d). \tag{3.24}$$

3.4 Experimental Setup

We modified Phan's GibbsLDA++ package[1] for the implementation of JST and Reverse-JST. The performance of the JST and Reverse-JST models are evaluated on two publicly available datasets as detailed in Section 3.4.1. Sections 3.4.2 and 3.4.3 describe the model prior and parameter settings, respectively, followed by document-level sentiment classification metric presented in Section 3.4.4.

3.4.1 Datasets

Two publicly available datasets, the movie review (MR)[2] and multi-domain sentiment (MDS) datasets[3], were used in our experiments.

Movie Review Dataset: The MR dataset becomes a benchmark for many studies in sentiment classification since the work of Pang et al. [2002]. Version 2.0 used in our

[1] http://gibbslda.sourceforge.net/

[2] http://www.cs.cornell.edu/people/pabo/movie-review-data

[3] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/index2.html

experiment consists of 1000 positive and 1000 negative movie reviews crawled from the IMDB movie archive, with an average of 30 sentences in each document. The sentiment label for each review is converted automatically from the author rating accompanying the review. For instance, with a five-star system (or compatible number systems), four stars and up are considered positive whereas two stars and below are considered negative. During the corpus construction, it was also imposed that only up to 20 reviews can be included for each review author per sentiment category.

Subjectivity MR Dataset: We experimented with another dataset, namely *subjective MR*, by removing the sentences that do not bear opinion information from the MR dataset, following the approach of Pang and Lee [2004]. The resulting dataset still contains 2000 documents with a total of 334,336 words and 18,013 distinct terms, about half the size of the original MR dataset without performing subjectivity detection.

Multi-domain Sentiment Dataset: First used by Blitzer et al. [2007], the MDS dataset contains 4 different types of product reviews crawled from Amazon.com including Book, DVD, Electronics and Kitchen, with 1000 positive and 1000 negative examples for each domain. Similar to the MR dataset, user rating accompanying each review was used for sentiment labelling, i.e., reviews with rating greater 3 were labeled positive, those with rating less than 3 were labeled negative, and the rest discarded because their polarity was ambiguous. We did not perform subjectivity detection on the MDS dataset because its average document length is much shorter than that of the MR dataset, with some documents even containing only a single sentence.

Preprocessing was performed on both of the datasets. Firstly, punctuation, numbers, non-alphabet characters and stop words were removed. Secondly, Porter stemming [Porter, 2006] was performed in order to reduce the vocabulary size and address the issue of data sparseness. Summary statistics of the datasets before and after preprocessing are shown in Table 3.2.

3.4.2 Defining Model Priors

In the experiments, two subjectivity lexicons, namely the MPQA[1] and the appraisal lexicons[2], were combined and incorporated as prior information into the model learning. These two lexicons contain lexical words whose polarity orientation has been fully specified. We extracted the words with strong positive and negative orientation and performed

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

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

Table 3.3: Prior information statistics.

Prior lexicon (pos./neg.)	MR	subjMR	Book
No. of distinct words	1,248/1,877	1,150/1,667	1,008/1,360
Total occurrence	108,576/57,744	67,751/34,276	31,697/14,006
Coverage (%)	17/9	20/10	13/6
Prior lexicon (pos./neg.)	DVD	Electronics	Kitchen
No. of distinct words	987/1320	571/555	595/514
Total occurrence	31,498/13,935	19,599/6,245	18,178/6,099
Coverage (%)	14/6	13/4	14/5

stemming in the preprocessing. In addition, words whose polarity changed after stemming were removed automatically, resulting in 1584 positive and 2612 negative words.

It is worth noting that the lexicons used here are fully domain-independent and do not bear any supervised information specific to the MR, subjMR and MDS datasets. Finally, the prior information was produced by retaining all words in the MPQA and appraisal lexicons that occurred in the experimental datasets. Statistics about the prior information for each dataset are listed in Table 3.3. It can be observed that the prior positive words occur much more frequently than the negative words, with frequencies at least doubling those of negative words in all of the datasets.

3.4.3 Hyperparameter Settings

Previous studies have shown that while LDA can produce reasonable results with a simple symmetric Dirichlet prior, an asymmetric prior over the document-topic distributions has substantial advantage over a symmetric prior [Wallach et al., 2009]. In the JST model implementation, we set the asymmetric prior $\boldsymbol{\beta} = 0.01$ in the initialization [Steyvers and Griffiths, 2007], and the symmetric prior $\gamma = (0.05 \times L)/S$, where L is the average document length, S the is total number of sentiment labels, and the value of 0.05 on average allocates 5% of probability mass for mixing. The asymmetric prior $\boldsymbol{\alpha}$ is learned directly from data using maximum-likelihood estimation [Minka, 2003] and updated every 25 iterations during the Gibbs sampling procedure. In terms of Reverse-JST, we set the asymmetric $\boldsymbol{\beta} = 0.01$, symmetry $\gamma = (0.05 \times L)/(T \times S)$, and the asymmetric prior $\boldsymbol{\alpha}$ is also learned from data as in JST.

3.4.4 Classifying Document Sentiment

The document sentiment is classified based on $P(\mathbf{l}|d)$, the probability of a sentiment label given document. In our experiments, we only consider the probability of positive and negative labels for a given document, with the neutral label probability being ignored. There are two reasons for this. First, sentiment classification for both the MR and MDS datasets is effectively a binary classification problem, i.e., documents are classified either as positive or negative, without the alternative of neutral. Second, the prior information we incorporated merely contributes to the positive and negative words, and consequently there will be much more influence on the probability distribution of positive and negative labels for a given document, rather than the distribution of neutral labels in the given document. Therefore, we define that a document d is classified as a positive-sentiment document if the probability of a positive sentiment label $P(l_{pos}|d)$ is greater than its probability of a negative sentiment label $P(l_{neg}|d)$, and vice versa.

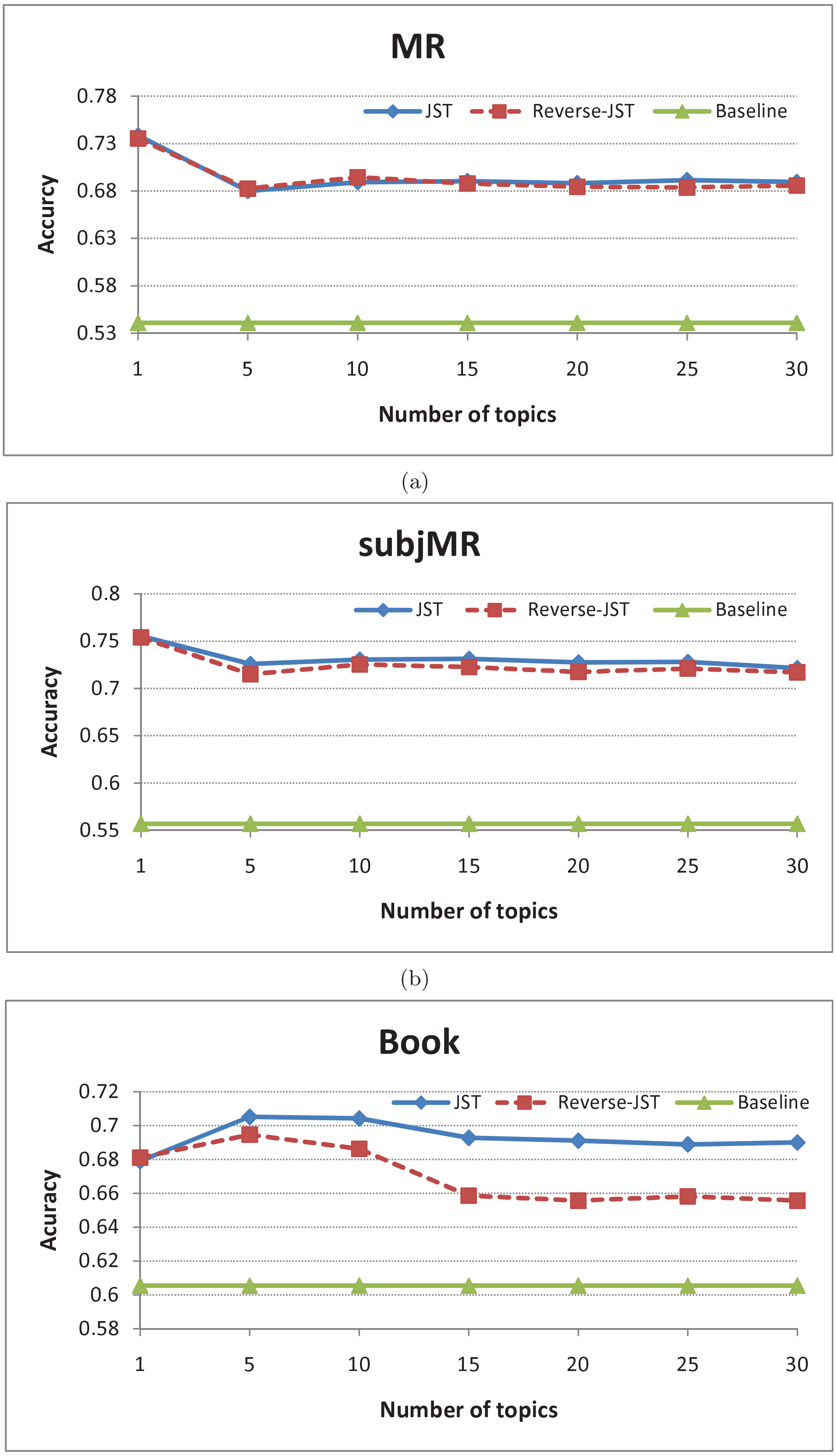
3.5 Experimental Results

In this section, we present and discuss the experimental results of both document-level sentiment classification and topic extraction based on the MR and MDS datasets.

3.5.1 Sentiment Classification Results vs. Different Number of Topics

As both JST and Reverse-JST model sentiment and topic mixtures simultaneously, it is worth exploring how the sentiment classification and topic extraction tasks affect/benefit each other and, in addition, how these two models behave with different topic number settings on different datasets when prior information is incorporated. With this in mind, we conducted a set of experiments on JST and Reverse-JST, with topic number $T \in \{1, 5, 10, 15, 20, 25, 30\}$. It is worth noting that as JST models the same number of topics under each sentiment label, with three sentiment labels, the total topic number of JST will be equivalent to a standard LDA model with $T \in \{3, 15, 30, 45, 60, 75, 90\}$.

Figure 3.3 shows the sentiment classification results of both JST and Reverse-JST at document level with prior information extracted from the MPQA and appraisal lexicons being incorporated. For all the reported results, accuracy is used as performance measure and the results were averaged over 10 runs. The baseline is calculated by counting the overlap of the prior lexicon with the training corpus. If the positive sentiment word count

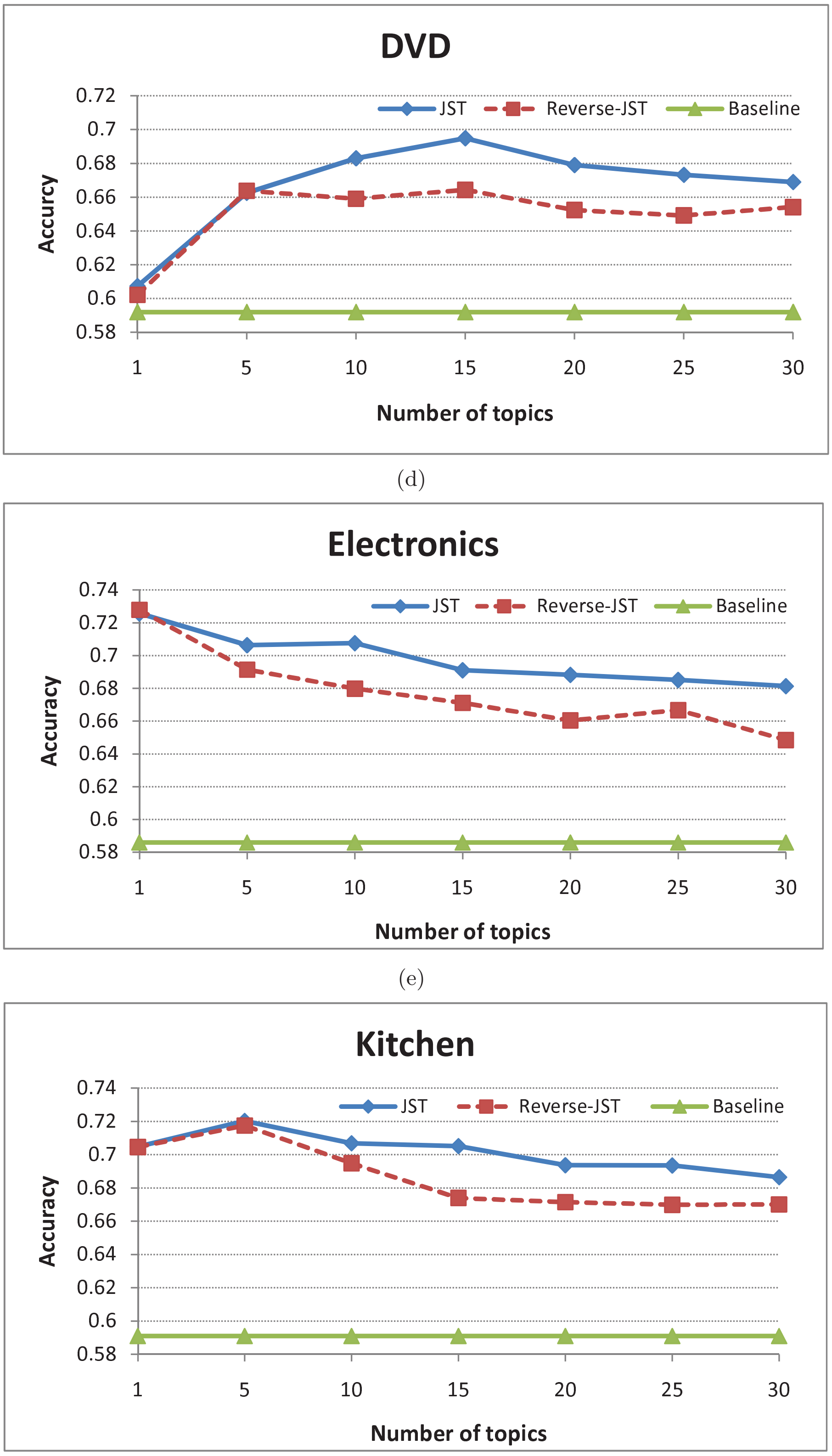


(a)

(b)

(c)

Figure 3.3: Sentiment classification accuracy vs. different topic number settings.



(d)

(e)

(f)

Figure 3.3: Sentiment classification accuracy vs. different topic number settings.

Table 3.4: Significant test results. Note: blank denotes the performance of JST and Reverse-JST is significantly undistinguishable; * denotes JST significantly outperforms Reverse-JST.

T	MR	subjMR	Book	DVD	Electronics	Kitchen
5						
10				*		
15			*	*	*	*
20			*	*	*	*
25	*		*	*	*	*
30			*	*	*	

is greater than that of the negative words, a document is classified as positive, and vice versa. The improvement over this baseline will reflect how much JST and Reverse-JST can learn from data.

As can be seen from Figure 3.3, both JST and Reverse-JST have a significant improvement over the baseline in all of the datasets. When the topic number is set to 1, both JST and Reverse-JST essentially become the standard LDA model with only three sentiment topics, and hence ignore the correlation between sentiment labels and topics. Figures 3.3c, 3.3d and 3.3f show that both JST and Reverse-JST perform better with multiple topic settings in the Book, DVD and Kitchen domains; especially noticeable is JST with 10% improvement at $T = 15$ over single topic setting on the DVD domain. This observation shows that modelling sentiment and topics simultaneously does indeed help improve sentiment classification. For the cases where a single topic performs best (i.e., Figure 3.3a, 3.3b and 3.3e), it is observed that apart from the MR dataset, the drop in sentiment classification accuracy by additionally modelling mixtures of topics is only marginal (i.e., 1% and 2% point drop in subjMR and Electronics, respectively), but both JST and Reverse-JST are able to extract sentiment-oriented topics in addition to document-level sentiment detection. Moreover, we proposed a mechanism for effectively incorporating prior information compared to the original LDA model.

When comparing JST with Reverse-JST, there are three main observations. First, JST outperforms Reverse-JST in most of the datasets with multiple topic settings, with up to 4% difference in the Book domain. Second, the performance difference between JST and Reverse-JST has some correlation with the corpus size (cf. Table 3.2). That is, when the corpus size is large these two models perform almost the same, e.g., on the MR dataset. In contrast, when the corpus size is relatively small JST significantly outperforms Reverse-JST, e.g., on the MDS dataset. It is also noticeable that in no case did Reverse-JST

Table 3.5: Performance comparison with existing models on sentiment classification accuracy. (Note: boldface denotes the best results.)

	Accuracy (%)						
	MR	subjMR	MDS				
			Book	DVD	Electronics	Kitchen	MDS overall
Baseline	54.1	55.7	60.6	59.2	58.6	59.1	59.4
JST	**73.9**	**75.6**	**70.5**	**69.5**	72.6	**72.1**	**71.2**
Reverse-JST	73.5	75.4	69.5	66.4	**72.8**	71.7	70.1
Dasgupta and Ng [2009b]	70.9	N/A	69.5	70.8	65.8	69.7	68.9
Li et al. [2009] (10% doc. label)	60.0	N/A	N/A				62.0
Li et al. [2009] (40% doc. label)	73.5	N/A					73.0

significantly outperform JST. A significance measure based on the paired t-test (critical $P = 0.05$) is reported in Table 3.4. Third, for both models, the sentiment classification accuracy is less affected by topic number settings when the dataset size is large. For instance, classification accuracy stays almost the same for the MR and subjMR datasets when topic number is increased from 5 to 30, whereas in contrast, a 2-3% drop is observed for Electronics and Kitchen. By closely examining the posterior of JST and Reverse-JST (cf. Equations 3.18 and 3.23), we noticed that the count $N_{d,j}$ (number of times topic j is associated with some word tokens in document d) in the Reverse-JST posterior would be relatively small due to the factor of a large topic number setting. On the contrary, the count $N_{d,k}$ (number of times sentiment label k is assigned to some word tokens in document d) in the JST posterior would be relatively large as k is only defined over 3 different sentiment labels. This essentially makes JST less sensitive to the data sparseness problem and to the perturbation of hyperparameter settings. In addition, JST encodes the assumption that there is approximately a single sentiment for the entire document, i.e., documents are mostly either positive or negative. This assumption is important as it allows the model to cluster different terms which share similar sentiment. In Reverse-JST, this assumption is not enforced unless only one topic for each sentiment label is defined. Therefore, JST appears to be a more appropriate model design for joint sentiment topic detection.

3.5.2 Comparison with Existing Models

In this section, we compare the overall sentiment classification performance of JST and Reverse-JST with some existing semi-supervised approaches [Dasgupta and Ng, 2009b; Li et al., 2009]. As can be seen from Table 3.5, the baseline results calculated based on the sentiment lexicon are below 60% for most of the datasets. By incorporating the same prior lexicon, a significant improvement is observed for JST and Reverse-JST over the baseline, where both models have over 20% performance gain on the MR and subjMR datasets, and 10-14% improvement on the MDS dataset. For the movie review data, there is a further 2% improvement for both models on the subjMR dataset over the original MR dataset. This suggests that although the subjMR dataset is in a much compressed form, it is more effective than the full dataset as it retains comparable polarity information in a much cleaner way [Pang and Lee, 2004]. In terms of the MDS dataset, both JST and Reverse-JST perform better on Electronics and Kitchen than Book and DVD, with about 2% difference in accuracy. Manually analysing the MDS dataset reveals that the Book and DVD reviews often contain a lot of descriptions of book contents or movie plots, which makes the reviews of these two domains difficult to classify; in contrast, in Electronics and Kitchen domains, opinions on products are often expressed in a much more straightforward manner. In terms of the overall performance, except in Electronics, it was observed that JST performed slightly better than Reverse-JST in all sets of experiments, with differences of 0.2-3% being observed.

When compared to the weakly-supervised approach based on a spectral clustering algorithm [Dasgupta and Ng, 2009b], except in the DVD domain where its accuracy is slightly lower, JST achieved better performance with more than 3% overall improvement. We point out that the proposed approach [Dasgupta and Ng, 2009b] requires users to specify which dimensions (defined by the eigenvectors in spectral clustering) are most closely related to sentiment by inspecting a set of features derived from the reviews for each dimension, and clustering is performed again on the data to derive the final results. In contrast, for the JST and Reverse-JST models proposed here, no human judgement is required. The non-negative matrix tri-factorization approach [Li et al., 2009] also employed lexical prior knowledge for semi-supervised sentiment classification. However, when incorporating 10% of labelled documents for training, the non-negative matrix tri-factorization approach performed much worse than JST, with only around 60% accuracy being achieved for all the datasets. Even with 40% labelled documents, it still performs worse than JST on the MR dataset and only slightly outperforms JST on the MDS dataset. It is worth noting that no labelled documents were used in the JST results reported here.

3.5.3 Sentiment Classification Results with Different Features

While JST and Reverse-JST models can give better or comparable performance in document-level sentiment classification compared to semi-supervised approaches [Dasgupta and Ng, 2009b; Li et al., 2009] with unigram features, it is worth considering the dependency

Table 3.6: Unigram and bigram features statistics.

Dataset	# of features (Unit: thousand)					
	MR	subjMR	MDS			
			Book	DVD	Electronics	Kitchen
unigrams	626	334	232	226	150	126
bigrams	1,239	680	318	307	201	170
unigrams+bigrams	1,865	1,014	550	533	351	296

Table 3.7: Sentiment classification results with different features. Note: boldface denotes the best results.

	Accuracy(%)					
	JST			Reverse-JST		
	unigrams	bigrams	unigrams+bigrams	unigrams	bigrams	unigrams+bigrams
MR	73.9	74.0	**76.6**	73.5	74.1	76.6
subjMR	75.6	75.6	**77.7**	75.4	75.5	77.6
Book	70.5	70.3	**70.8**	69.5	69.7	69.8
DVD	69.5	71.3	**72.5**	66.4	71.4	72.4
Electronics	72.6	70.2	74.9	72.8	70.5	**75.0**
Kitchen	**72.1**	70.0	70.8	71.7	69.9	70.5

between words since it might serve an important function in sentiment analysis. For instance, phrases expressing negative sentiment such as "*not good*" or "*not durable*" will convey completely different polarity meanings without considering negations. Therefore, we extended the JST and Reverse-JST models to include higher order information, i.e., bigrams, for model learning. Table 3.6 shows the feature statistics of the datasets in unigrams, bigrams and the combination of both. For the negator lexicon, we collect a handful of words from the General Inquirer under the NotLw category[1]. We experimented with topic number $T \in \{1, 5, 10, 15, 20, 25, 30\}$. However, it was found that JST and Reverse-JST achieved best results with single topic on bigrams and the combination of bigrams and unigrams most of the time, except for a few cases where multiple topics performed better (i.e., JST and Reverse-JST with $T = 5$ on Book using unigrams+bigrams, as well as Reverse-JST with $T = 10$ on Electronics using unigrams+bigrams). This is probably due to the fact that bigram features have much lower frequency counts than unigrams. Thus, with the sparse feature co-occurrence, multiple topic settings likely fail to cluster different terms that share similar sentiment and hence harm the sentiment classification accuracy.

Table 3.7 shows the sentiment classification results of JST and Reverse-JST using different features. It can be observed that both JST and Reverse-JST perform almost the

[1] `http://www.wjh.harvard.edu/~inquirer/NotLw.html`

Table 3.8: Topic examples extracted by JST under different sentiment labels.

	MR		Book		DVD		Electronics		Kitchen	
Positive sentiment label	ship	good	recip	children	action	funni	mous	sound	color	recommend
	titan	realli	food	learn	good	comedi	hand	qualiti	beauti	highli
	crew	plai	cook	school	fight	make	logitech	stereo	plate	impress
	cameron	great	cookbook	child	right	humor	comfort	good	durabl	love
	alien	just	beauti	ag	scene	laugh	scroll	high	qualiti	favorit
	jack	perform	simpl	parent	chase	charact	wheel	listen	fiestawar	especi
	water	nice	eat	student	hit	joke	smooth	volum	blue	nice
	stori	fun	famili	teach	art	peter	feel	decent	finger	beautifulli
	rise	lot	ic	colleg	martial	allen	accur	music	white	absolut
	rose	act	kitchen	think	stunt	entertain	track	hear	dinnerwar	fabul
	boat	direct	varieti	young	chan	funniest	touch	audio	bright	bargin
	deep	best	good	cours	brilliant	sweet	click	set	purpl	valu
	ocean	get	pictur	educ	hero	constantli	conveni	price	scarlet	excel
	dicaprio	entertain	tast	kid	style	accent	month	speaker	dark	bought
	sink	better	cream	english	chines	happi	mice	level	eleg	solid
Negative sentiment label	prison	bad	polit	war	horror	murder	drive	batteri	fan	amazon
	evil	worst	east	militari	scari	killer	fail	charg	room	order
	guard	plot	middl	armi	bad	cop	data	old	cool	return
	green	stupid	islam	soldier	evil	crime	complet	life	air	ship
	hank	act	unit	govern	dead	case	lose	unaccept	loud	sent
	wonder	suppos	inconsist	thing	blood	prison	failur	charger	nois	refund
	excute	script	democrat	evid	monster	detect	recogn	period	live	receiv
	secret	wast	influenc	led	zombi	investig	backup	longer	annoi	damag
	mile	dialogu	politician	iraq	fear	mysteri	poorli	recharg	blow	dissapoint
	death	bore	disput	polici	scare	commit	error	hour	vornado	websit
	base	poor	cultur	destruct	live	thriller	storag	last	bedroom	discount
	tom	complet	eastern	critic	ghost	attornei	gb	power	inferior	polici
	convict	line	polici	inspect	devil	undercov	flash	bui	window	unhappi
	return	terribl	state	invas	head	suspect	disast	worthless	vibrat	badli
	franklin	mess	understand	court	creepi	shock	yesterdai	realli	power	shouldn

same with unigrams or bigrams on the MR, subjMR, and Book datasets. However, using bigrams gives a better accuracy in DVD, but is worse on Electronics and Kitchen compared to using unigrams for both models. When combining both unigrams and bigrams, a performance gain is observed for most of the datasets except the Kitchen data. For both MR and subjMR, using the combination of unigrams and bigrams gives more than 2% improvement compared to using either unigrams or bigrams alone, with 76.6% and 77.7% accuracy being achieved on these two datasets, respectively. For the MDS dataset, the combined features slightly outperform unigrams and bigrams on Book and give a significant gain on DVD (i.e., 3% over unigrams; 1.2% over bigrams) and Electronics (i.e., 2.3% over unigrams; 4.7% over bigrams). Thus, we may conclude that the combination of unigrams and bigrams gives the best overall performance.

3.5.4 Topic Extraction

The second goal of JST is to extract topics from the MR (without subjectivity detection) and MDS datasets, and evaluate the effectiveness of topic sentiment captured by the model. Unlike the LDA model where a word is drawn from the topic-word distribution, in JST one draws a word from the per-corpus word distribution conditioned on both topics and sentiment labels. Therefore, we analyse the extracted topics under positive and negative sentiment labels separately. 20 topic examples extracted from the MR and MDS datasets are shown in Table 3.8, where each topic was drawn from a particular domain under a sentiment label.

Topics in the top half of Table 3.8 were generated under the positive sentiment label and the remaining topics were generated under the negative sentiment label, each of which is represented by the top 15 topic words. As can be seen from the table, the extracted topics are quite informative and coherent. The movie review topics appear to capture the underlying theme of a movie or the relevant comments from a movie reviewer, while the topics from the MDS dataset represent a certain product review from the corresponding domain. For example, for the two positive sentiment topics under the movie review domain, the first is closely related to the very popular romantic movie *"Titanic"* directed by James Cameron and starring by Leonardo DiCaprio and Kate Winslet, whereas the other one is likely to be a positive review for a movie. Regarding the MDS dataset, the first topics for Book and DVD under the positive sentiment label probably discuss a good cookbook and a popular action movie by Jackie Chan, respectively; for the first negative topic of Electronics, it is likely to be about complaints regarding data loss due to a flash drive failure, while the first negative topic of the kitchen domain is probably about the dissatisfaction with the high noise level of the *Vornado* brand fan.

In terms of topic sentiment, by examining each of the topics in Table 3.8, it is quite evident that most of the positive and negative topics indeed bear positive and negative sentiment. The first movie review topic and the second Book topic under the positive sentiment label mainly describe movie plot and the contents of a book, with fewer words carrying positive sentiment compared to other positive sentiment topics under the same domain. Manually examining the data reveals that the terms that seem not to convey sentiments under those two topics in fact appear in the context of expressing positive sentiments. Overall, the above analysis illustrates the effectiveness of JST in extracting opinionated topics from a corpus.

3.6 Discussion

In this chapter, we have presented a joint sentiment-topic (JST) model and a reparameterized version of JST called Reverse-JST. While most of the existing approaches to sentiment classification favour supervised learning, both the JST and Reverse-JST models target sentiment and topic detection simultaneously in a weakly-supervised fashion. Without a hierarchical prior, JST and Reverse-JST are essentially equivalent. However, extensive experiments conducted on datasets across different domains reveal that these two models behave very differently when sentiment prior knowledge is incorporated, in which case JST consistently outperformed Reverse-JST. For general domain sentiment classification, by incorporating a small amount of domain-independent prior knowledge, the JST model achieved either better or comparable performance compared to existing semi-supervised approaches despite using no labelled documents, which demonstrates the flexibility of JST in the sentiment classification task. Moreover, the topics and topic sentiments detected by JST are indeed coherent and informative.

The JST model described in this chapter is still a static model which does not model topic and sentiment dynamics. In order to facilitate the demand of modeling large amounts of user-generated data with both topic and sentiment distributions that evolve over time, we present a dynamic version of the JST model called dJST in the next chapter, which allows the detection and tracking of views of current and recurrent interests and shifts in topic and sentiment.

Chapter 4

Dynamic Joint Sentiment-Topic Model

4.1 Introduction

In this chapter we present a new member of the probabilistic dynamic topic models called the dynamic joint sentiment-topic (dJST) model for sequentially analysing the topic and sentiment evolution over time in a collection of documents. The development of the dJST model is motivated by two observations. First, many large datasets are temporally dependent, where the pattern of the documents collected in the early stage may not be preserved for the documents collected in later stages. The previously proposed JST model [Lin et al., 2011b] assumes that words in the documents have a static co-occurrence pattern, which may not be suitable for the task of capturing topic and sentiment shifts in a time-variant corpus. Second, when fitting a large scale dataset, 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 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. As the past data are not required for inference, dJST can easily analyse massive document collections without the need to store any documents locally. 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 modelled 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 expectation-maximization (EM) using the fixed-point iteration method [Minka, 2003].

We compared the performance of the dJST model with two non-dynamic versions of JST, namely, JST-one which uses only the data in the last epoch for training, and JST-all which uses all the past data for model learning. Experimental results on the Mozilla add-on review dataset which we crawled from 2007 to 2011 show that dJST outperforms JST-one in both predictive perplexity and sentiment classification accuracy, which demonstrates the effectiveness of modelling dynamics. Detailed discussion of the dataset is given in Section 4.3.1. On the other hand, while JST-all achieves slightly better sentiment classification accuracy than dJST, the perplexity of dJST is much lower. Besides, by avoiding the modelling of all the past documents as JST-all, the computational time of dJST is just in a fraction of JST-all.

The rest of the chapter is organized as follows. We first describe the dynamic JST model and the online inference algorithm in Section 4.2. Experimental setup and results based on the Mozilla review dataset are then presented in Section 4.3 and Section 4.4, respectively. Finally, we conclude the work in Section 4.5.

4.2 Dynamic JST (dJST) Model

In a time-stamped document collection, we assume documents are received as a stream in a series of epochs, where an epoch is a period which can be set at arbitrary time granularity, e.g., an hour, a day, or a year and the documents $\{d_1^t, \cdots, d_D^t\}$ received within epoch t are sorted in the ascending order of their time stamps. Also, a document d at epoch t

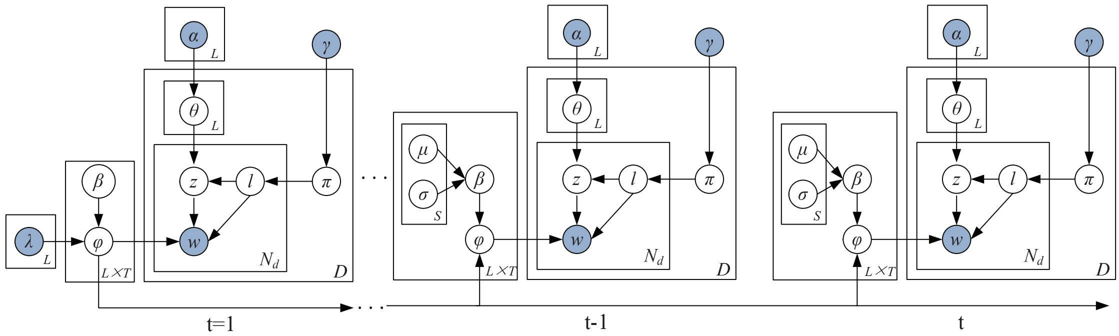


Figure 4.1: Dynamic JST model.

is represented as a vector of word tokens $\mathbf{w}_d^t = (w_{d_1}^t, w_{d_2}^t, \cdots, w_{d_{N_d}}^t)$, where the bold-font variables denote vectors.

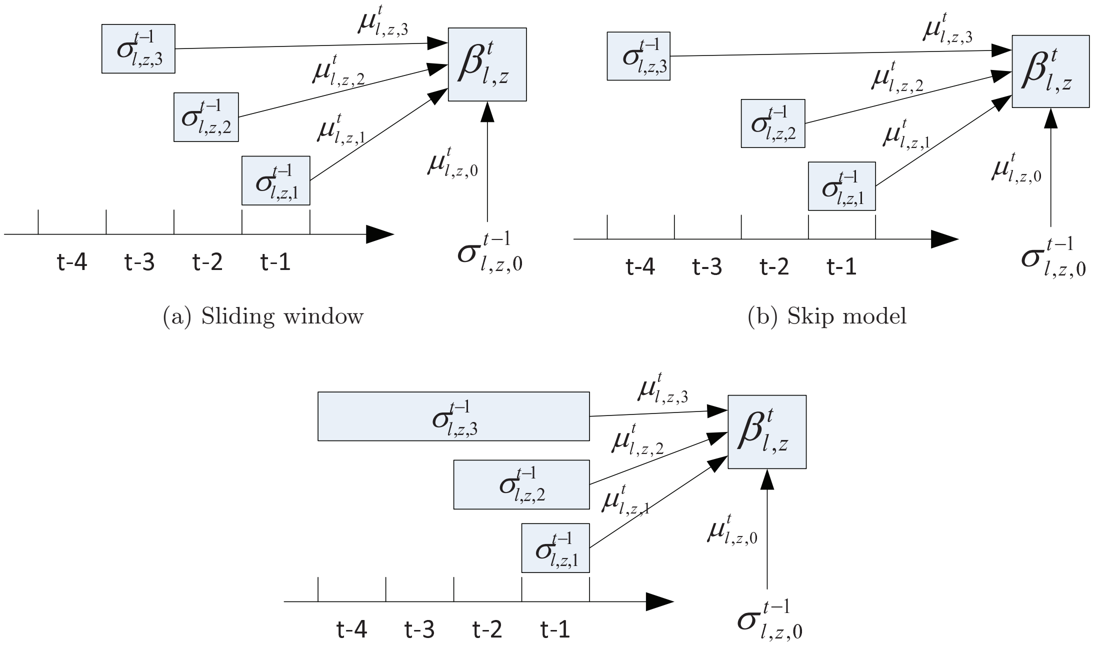
When fitting such a time-stamped document collection using the dJST model as shown in Figure 4.1, it is assumed that the document generation in the current epoch is influenced by the documents from past epochs. In order to account for the historical dependencies at different time scales, we introduced an evolutionary matrix $\mathbf{E}_{l,z}^{t-1} = \{\boldsymbol{\sigma}_{l,z,s}^{t-1}\}_{s=0}^{S}$ to dJST. Here $\mathbf{E}_{l,z}^{t-1}$ is the word distributions at previous epochs $t-1$ for topic z, sentiment label l, and for each time slice s; whereas $\boldsymbol{\sigma}_{l,z,s}^{t-1} = \{\sigma_{l,z,s,w}^{t-1}\}_{w=1}^{V}$ is the column vector of $\mathbf{E}_{l,z}^{t-1}$ which represents the word distribution of topic z and sentiment label l for the documents received within the time slice specified by s.

One intrinsic difference between the original JST model and the dJST model lies in that in JST the sentiment-topic word distributions $\boldsymbol{\varphi}_{l,z}$ are generated with a symmetric Dirichlet prior modified by a transformation matrix λ; whereas in dJST the current sentiment-topic word distributions $\boldsymbol{\varphi}_{l,z}^t$ at epoch t are generated from the Dirichlet distribution parameterized by $\mathbf{E}_{l,z}^{t-1}$. 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, which can be written as

$$\boldsymbol{\varphi}_{l,z}^t \sim \mathrm{Dir}(\boldsymbol{\beta}_{l,z}^t), \tag{4.1}$$

$$\boldsymbol{\beta}_{l,z}^t = \boldsymbol{\mu}_{l,z}^t \mathbf{E}_{l,z}^{t-1}, \tag{4.2}$$

where $\boldsymbol{\beta}_{l,z}^t$ is the Dirichlet prior for the sentiment-topic word distributions at epoch t, and $\boldsymbol{\mu}_{l,z}^t = \{\mu_{l,z,s}^t\}_{s=0}^{S}$ is a $S+1$ dimensional weight vector ($\mu_{l,z,s}^t > 0, \sum_{s=0}^{S} \mu_{l,z,s}^t = 1$) with its components representing the weights that each time slice s contributes to calculating



(a) Sliding window

(b) Skip model

(c) Multiscale model

Figure 4.2: Three different time slice models with a total number of historical time slices S=3. The variable $t-1$ denotes epoch $t-1$.

the priors of $\varphi^t_{l,z}$. 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 historical context can be encoded into the evolutionary matrix $\mathbf{E}^{t-1}_{l,z}$ in many possible ways. In our experiments, we have explored three different time slice settings which are listed below:

- *Sliding window.* If $s \in \{t-S, t-S+1, \cdots, t-1\}$, 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 $s \in \{t-2^{S-1}, t-2^{S-2} \cdots, t-1\}$, this takes historical sentiment-topic-word distributions into account by skipping some epochs. 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 can also account for the influence of the past at different timescales to the current epoch [Iwata et al., 2010; Nallapati et al., 2007]. For example, we can set time slice s equivalent to 2^{s-1} epochs. Hence, if $S = 3$, we 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 the taking into consideration of previous long- and short- timescale distributions. However, this model would take more time and memory and hence effective approximation needs to be performed in order to reduce the time and memory costs.

For a better illustration, we show in Figure 4.2 the three time slice settings for dJST with a total number of historical time slices $S = 3$.

When modelling historical dependencies, one effect is that the influence of the dependencies may vary over time. Therefore, in order for dJST to flexibly respond to the influences of the historical dependencies on the current epoch, it is important to estimate the weight vector $\boldsymbol{\mu}_{l,z}^{t}$ effectively. We give a detailed discussion of estimating $\boldsymbol{\mu}_{l,z}^{t}$ in Section 4.2.2.

The notations used for the dJST model are summarized in Table 4.1.

Finally, assuming we have already calculated the evolutionary parameters $\{\mathbf{E}_{l,z}^{t-1}, \boldsymbol{\mu}_{l,z}^{t}\}$ for the current epoch t, the generative story for the dJST model as shown in Figure 4.1 at epoch t is as follows:

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

Table 4.1: Parameter notations for the dJST model.

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
S	number of time slices
$\boldsymbol{\alpha}^t$	asymmetric Dirichlet priors on the mixing topic proportions at epoch t, $\boldsymbol{\alpha}^t = \{\{\alpha_{l,z}\}_{z=1}^T\}_{l=1}^S$ ($S \times T$ matrix)
$\boldsymbol{\beta}^t$	asymmetric Dirichlet priors on the sentiment label and topic specific word distribution at epoch t, $\boldsymbol{\beta}^t = \{\{\{\beta_{l,z,w}\}_{w=1}^V\}_{z=1}^T\}_{l=1}^L$ ($L \times T \times V$ matrix).
γ^t	symmetric Dirichlet priors on the mixing sentiment proportions at epoch t (scalar).
$\boldsymbol{\pi}_d^t$	parameter notation for the sentiment mixing proportions for document d at epoch t ($L-$ vector). For D^t documents, $\boldsymbol{\Pi}^t = \{\{\pi_{d,l}^t\}_{l=1}^L\}_{d=1}^{D^t}$ ($D^t \times L$ matrix).
$\boldsymbol{\theta}_{d,l}^t$	parameter notation for the topic mixing proportions for document d and sentiment label l at epoch t ($T-$ vector). For D^t documents and L sentiment labels, $\boldsymbol{\Theta}^t = \{\{\{\theta_{d,l,z}^t\}_{z=1}^T\}_{l=1}^L\}_{d=1}^{D^t}$ ($D^t \times L \times T$ matrix).
$\boldsymbol{\varphi}_{l,z}^t$	parameter notation for the multinomial distribution over words for sentiment label l and topic z at epoch t ($V-$ vector). For L sentiment labels and T topics, $\boldsymbol{\Phi}^t = \{\{\{\varphi_{l,z,w}^t\}_{w=1}^V\}_{z=1}^T\}_{l=1}^L$ ($L \times T \times V$ matrix)
$\boldsymbol{\lambda}$	parameter notation for the transformation matrix for encoding prior information ($L \times V$ matrix).
$\mathbf{E}_{l,z}^t$	evolutionary matrix of sentiment label l and topic z and S time slices at epoch t, $\mathbf{E}_{l,z}^t = \{\boldsymbol{\sigma}_{l,z,s}^t\}_{s=0}^S$ ($S \times V$ matrix). For L sentiment labels and T topics, $\mathbf{E}^t = \{\{\mathbf{E}_{l,z}^t\}_{z=1}^T\}_{l=1}^L$ ($L \times T \times S \times V$ matrix)
$\boldsymbol{\sigma}_{l,z,s}^t$	multinomial parameters for the word distribution of sentiment label l and topic z with time slice s at epoch t, $\boldsymbol{\sigma}_{l,z,s}^t = \{\sigma_{l,z,s,w}^t\}_{w=1}^V$ ($V-$ vector)
$\boldsymbol{\mu}_{l,z}^t$	$\boldsymbol{\mu}_{l,z}^t = \{\mu_{l,z,s}^t\}_{s=0}^S$ ($S+1$ vector). Each component determines the contribution of the corresponding time slice s in computing the priors for $\boldsymbol{\varphi}_{l,z}^t$

4.2.1 Online Inference

We present an online stochastic EM algorithm to sequentially update the dJST model parameters at each epoch using the newly received documents and the model estimated at the previous epoch. At each EM iteration, we iterate between inferring the latent sentiment labels and topics for the observed data using the collapsed Gibbs sampling algorithm and estimating the hyperparameters using maximum likelihood.

4.2.1.1 Deriving the Gibbs Sampler

The main objective of the inference in dJST is to find a set of model parameters $\{\mathbf{\Pi}^t$, $\mathbf{\Theta}^t$ and $\mathbf{\Phi}^t\}$ that best explain the newly obtained data. Here the posterior distributions of interest revealing the latent semantic structure of the data is intractable and we use the Gibbs sampling algorithm for approximate inference.

Analogous to JST, we can derive the Gibbs sampler for dJST by evaluating the joint distribution of the model. Given the evolutionary parameters $\mathbf{E}^{t-1}, \boldsymbol{\mu}^t$, the total probability of the current dJST model for the document collection $\mathbf{w}^t$ and the corresponding sentiment label $\mathbf{l}^t$ and topic $\mathbf{z}^t$ assignments at epoch t can be factorized into three terms

$$P(\mathbf{w}^t, \mathbf{l}^t, \mathbf{z}^t | \mathbf{E}^{t-1}, \boldsymbol{\mu}^t, \boldsymbol{\alpha}^t, \gamma^t) = P(\mathbf{l}^t|\gamma^t) P(\mathbf{z}^t|\mathbf{l}^t, \boldsymbol{\alpha}^t) P(\mathbf{w}^t|\mathbf{l}^t, \mathbf{z}^t, \mathbf{E}^{t-1}, \boldsymbol{\mu}^t), \qquad (4.3)$$

where each term on the right hand side (RHS) can be handled separately because each of the model parameters $\mathbf{\Pi}^t$, $\mathbf{\Theta}^t$ and $\mathbf{\Phi}^t$ appears in only one of the terms. Here we do not give the full derivation of the dJST Gibbs sampler as its derivation it is very similar to that of the JST model. For the details of the full derivation, please refer to Section 3.2.2.

For the first term on the RHS of Equation 4.3, by integrating out $\mathbf{\Pi}^t$, we obtain

$$P(\mathbf{l}^t|\gamma^t) = \prod_{d=1}^{D^t} \frac{\Gamma(L\gamma^t)}{\Gamma(\gamma^t)^L} \frac{\prod_{l=1}^{L} \Gamma(N_{d,l}^t + \gamma^t)}{\Gamma(N_d^t + L\gamma^t)}, \qquad (4.4)$$

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

For the second term, by integrating out $\mathbf{\Theta}^t$, we obtain

$$P(\mathbf{z}^t|\mathbf{l}^t, \boldsymbol{\alpha}^t) = \prod_{d=1}^{D^t} \prod_{l=1}^{L} \frac{\Gamma(\sum_{z=1}^{T} \alpha_{l,z}^t)}{\prod_{z=1}^{T} \Gamma(\alpha_{l,z}^t)} \frac{\prod_z \Gamma(N_{d,l,z}^t + \alpha_{l,z}^t)}{\Gamma(N_{d,l}^t + \sum_z \alpha_{l,z}^t)}, \qquad (4.5)$$

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

For the third term, by using Equation 4.2 and integrating out $\mathbf{\Phi}^t$, we obtain

$$P(\mathbf{w}^t|\mathbf{l}^t, \mathbf{z}^t, \mathbf{E}^{t-1}, \boldsymbol{\mu}^t) = \prod_{l=1}^{L}\prod_{z=1}^{T} \frac{\Gamma(\sum_s \mu^t_{l,z,s})}{\prod_{w=1}^{V}\Gamma(\sum_s \mu^t_{l,z,s}\sigma^{t-1}_{l,z,s,w})} \frac{\prod_{w=1}^{V}\Gamma(N^t_{l,z,w} + \sum_s \mu^t_{l,z,s}\sigma^{t-1}_{l,z,s,w})}{\Gamma(N^t_{l,z} + \sum_s \mu^t_{l,z,s})}, \tag{4.6}$$

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

The full conditional distribution of dJST can then be derived from the joint distribution, from which the Gibbs sampler sequentially samples the hidden variables (here l^t and z^t) for each word token, given the current values of all other variables and data. Letting the index of a token be $x = (d, n, t)$ and the subscript $\neg x$ denote a quantity that excludes counts of word w_x in the nth position of document d at epoch t, the full conditional distribution for the sentiment label l^t and topic z^t assignments for w_x, by marginalizing out the random variables $\mathbf{\Pi}^t$, $\mathbf{\Theta}^t$, and $\mathbf{\Phi}^t$ is

$$P(z_x = j, l_x = k|\mathbf{w}^t, \mathbf{z}^t_{\neg x}, \mathbf{l}^t_{\neg x}, \mathbf{E}^{t-1}, \boldsymbol{\mu}^t) \propto \frac{N^t_{k,j,w_j,\neg x} + \sum_s \mu^t_{k,j,s}\sigma^{t-1}_{k,j,s,w_j}}{N^t_{k,j,\neg x} + \sum_s \mu^t_{k,j,s}} \cdot \frac{N^t_{d,k,j,\neg x} + \alpha^t_{k,j}}{N^t_{d,k,\neg x} + \sum_j \alpha^t_{k,j}} \cdot \frac{N^t_{d,k,\neg x} + \gamma^t}{N^t_{d\neg x} + L\gamma^t}. \tag{4.7}$$

Using Equation 4.7, the Gibbs sampling procedure can be run and samples obtained from the Markov chain are then used to estimate the dJST model parameters.

The approximate sentiment-topic word distribution at epoch t is

$$\varphi^t_{k,j,i} = \frac{N^t_{k,j,i} + \sum_s \mu^t_{k,j,s}\sigma^{t-1}_{k,j,s,i}}{N^t_{k,j} + \sum_s \mu^t_{k,j,s}}. \tag{4.8}$$

The approximate per-document sentiment label specific topic proportion at epoch t is

$$\theta^t_{d,k,j} = \frac{N^t_{d,k,j} + \alpha^t_{k,j}}{N^t_{d,k} + \sum_j \alpha^t_{k,j}}. \tag{4.9}$$

Finally, the approximate per-document sentiment proportion at epoch t is

$$\pi^t_{d,k} = \frac{N^t_{d,k} + \gamma^t}{N^t_d + L\gamma^t}. \tag{4.10}$$

4.2.2 Evolutionary Parameters Estimation

There are two sets of evolutionary parameters which need to be estimated in dJST: the evolutionary matrix $\mathbf{E}^t$ and the weight vector $\boldsymbol{\mu}^t$. The evolutionary matrix $\mathbf{E}^t$ is updated at the point when the Gibbs sampling procedure for the current epoch is completed. The weight vector $\boldsymbol{\mu}^t$ can either be estimated by maximizing the joint distribution in Equation 4.3 using the fixed-point iteration method [Minka, 2003] every 40 Gibbs sampling iterations, or set in the initialization with an exponential decay function.

4.2.2.1 Estimating the Evolutionary Matrix $\mathbf{E}^t$

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

$$\sigma^t_{l,z,s,w} = \frac{C^t_{l,z,s,w}}{\sum_w C^t_{l,z,s,w}}, \tag{4.11}$$

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

$$\hat{N}^{t'}_{l,z,w} = N^{t'}_{l,z,w}\varphi^{t'}_{l,z,w}, \tag{4.12}$$

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 associated with sentiment label l and topic z at epoch t' recovered using Equation 4.8. In contrast, 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:

$$C^t_{l,z,s,w} = \begin{cases} \hat{N}^{t'=t-s+1}_{l,z,w} & \text{Sliding window} \\ \hat{N}^{t'=t-2^s+1}_{l,z,w} & \text{Skip model} \\ \sum_{t'=t-2^{s-1}+1}^{t} \hat{N}^{t'}_{l,z,w} & \text{Multi-scale model} \end{cases} \tag{4.13}$$

Updating $C^t_{l,z,w,s}$ for the *Multi-scale model* requires 2^{s-1} additions. The difference between $C^t_{l,z,w,s}$ and $C^{t-1}_{l,z,s,w}$ is that the count of the earliest epoch in $C^{t-1}_{l,z,w,s}$ is replaced by the count of the latest epoch in $C^t_{l,z,w,s}$, thus the value of $C^t_{l,z,s,w}$ can be sequentially updated in a much more efficient way with just two additions :

$$C^t_{l,z,s,w} = C^{t-1}_{l,z,s,w} - \hat{N}^{t-2^{s-1}}_{l,z,w} + \hat{N}^t_{l,z,w} \tag{4.14}$$

With the *Multi-scale model*, the memory requirement increases exponentially with the number of time slices. Following Iwata et al. [2010], we approximate the update by reducing the frequency for updating the long-timescale frequencies, so that $C^t_{l,z,s,w}$ will only be updated if $t \bmod 2^{s-1} = 0$. This ensures that the memory requirement is linear in the number of time slices.

4.2.2.2 Estimating the Weight Vector $\boldsymbol{\mu}^t$

$\boldsymbol{\mu}^t$ is an influential weight vector which determines the contributions of each component of $\mathbf{E}^{t-1}$ in computing the Dirichlet priors on the word distributions $\boldsymbol{\Phi}^t$ for the current epoch t. Therefore, estimating $\boldsymbol{\mu}^t$ is important for dJST to flexibly handle the influence of the historical dependencies. 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 When accounting for the distributions of the documents from the past, a common observation is that the more recent documents would have a relatively stronger influence on the model parameters in the current epoch compared to the earlier documents. Such an effect can be modelled by an exponential decay function

$$\mu^t_s = \exp(-\kappa s), \tag{4.15}$$

where κ is the decay rate and s is a specific time slice. Figure 4.3 illustrates μ^t_s undergoing exponential decay with different decay rate settings for $s \in [0, 5]$. In our experiments, we empirically set $\kappa = 0.5$, and the weights for each sentiment label and each topic under the same time slice s are set with the same value, i.e., $\{\{\mu^t_{l,z,s}\}^T_{z=1}\}^L_{l=1} = \mu^t_s$.

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 4.3 using the fixed-point iteration method [Minka, 2003]. The update formula is:

$$(\mu^t_{l,z,s})^{\text{new}} \leftarrow \frac{\mu^t_{l,z,s} \sum_w \sigma^t_{l,z,s,w} A}{B}, \tag{4.16}$$

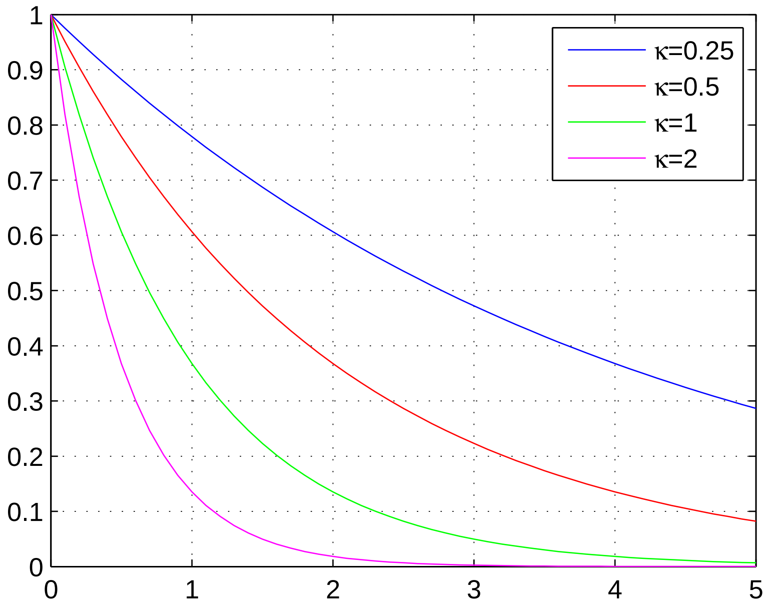


Figure 4.3: Exponential decay function with different decay rates.

where

$$A = \Psi(N^t_{l,z,w} + \sum_{s'} \mu^t_{l,z,s'}\sigma^t_{l,z,s',w}) - \Psi(\sum_{s'} \mu^t_{l,z,s'}\sigma^t_{l,z,s',w}), \tag{4.17}$$

$$B = \Psi(N^t_{l,z} + \sum_{s'} \mu^t_{l,z,s'}) - \Psi(\sum_{s'} \mu^t_{l,z,s'}), \tag{4.18}$$

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

The full derivation of Equation 4.16 is given in appendix B.

4.2.3 Hyperparameter Settings

For the dJST model hyperparameters, while the values of $\boldsymbol{\beta}^t$ and γ^t are set empirically, $\boldsymbol{\alpha}^t$ is estimated from data using maximum-likelihood as part of the online stochastic EM algorithm.

Setting $\boldsymbol{\alpha}^t$ A common practice for topic model implementation is to use symmetric Dirichlet hyperparameters. However, it was reported that using 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 $\boldsymbol{\alpha}^t = (0.05 \times \bar{N}^t)/(L \times T)$, where $\bar{N}^t$ is the average document length of epoch t and the

Algorithm 2 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
    $D^t = \{d_1^t, \cdots, d_M^t\}$
Output: Dynamic JST model
Sort documents according to their time stamps
for $t = 1$ to maxEpochs do
  if $t == 1$ then
    Set $\boldsymbol{\beta}^t = \boldsymbol{\lambda} \times \mathbf{0.01}$
  else
    Set $\boldsymbol{\mu}_{l,z}^t = 1/S$
    Set $\boldsymbol{\beta}_{l,z}^t = \boldsymbol{\mu}_{l,z}^t \mathbf{E}_{l,z}^{t-1}$
  end if
  Set $\boldsymbol{\alpha}^t = (0.05 \times \bar{N}^t)/(L \times T)$
  Set $\gamma^t = (0.05 \times \bar{N}^t)/L$
  Initialize $\boldsymbol{\Pi}^t, \boldsymbol{\Theta}^t, \boldsymbol{\Phi}^t$, and all count variables
  Initialize the sentiment label and topic assignment randomly for all word tokens in
  $D^t$
  for $i = 1$ to $max$ Gibbs Sampling Iterations do
    $[\mathbf{l}^t, \mathbf{z}^t]$ = GibbsSampling($D^t, \boldsymbol{\alpha}^t, \boldsymbol{\beta}^t, \gamma^t$)
    for every 40 Gibbs sampling iterations do
      Update $\boldsymbol{\alpha}^t$ using Equation 4.19
      Update $\boldsymbol{\mu}_{l,z}^t$ using Equation 4.15 or 4.16
      Update $\boldsymbol{\beta}_{l,z}^t = \boldsymbol{\mu}_{l,z}^t \mathbf{E}_{l,z}^{t-1}$
    end for
    for every 200 Gibbs sampling iterations do
      Update $\boldsymbol{\Pi}^t, \boldsymbol{\Theta}^t, \boldsymbol{\Phi}^t$ with the new sampling results
    end for
  end for
  Update $\mathbf{E}_{l,z}^{t-1}$ using Equation 4.11
end for
```

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; Wallach et al., 2009]:

$$(\alpha_{l,z}^t)^{\text{new}} \leftarrow \frac{\alpha_{l,z}^t \sum_d [\Psi(N_{d,l,z}^t + \alpha_{l,z}^t) - \Psi(\alpha_{l,z}^t)]}{\sum_d [\Psi(N_{d,l}^t + \sum_{z'} \alpha_{l,z'}^t) - \Psi(\sum_{z'} \alpha_{l,z'}^t)]}. \tag{4.19}$$

Setting $\boldsymbol{\beta}^t$ At the first epoch, the Dirichlet prior $\boldsymbol{\beta}^t$ is first initialized with symmetric value 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 following the procedure described in Chapter 3.2.1. Here, the sentiment lexicon being incorporated into dJST is identical to the one described in Section 3.4.2. For the subsequent epochs, if there are new words encountered (i.e., words that do not appear in the previous epoch), the word prior polarity information will be incorporated in a similar way for those new words. For the words that have appeared in the previous epoch, their Dirichlet priors for the sentiment-topic word distributions are calculated using Equation 4.2.

Setting γ^t We empirically set the symmetric prior $\gamma = (0.05 \times \bar{N}^t)/L$, 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 are summarized in Algorithm 2, which iterates between Gibbs sampling using Equation 4.7 and maximum likelihood estimation using Equations 4.16 and 4.19.

4.3 Experimental Setup

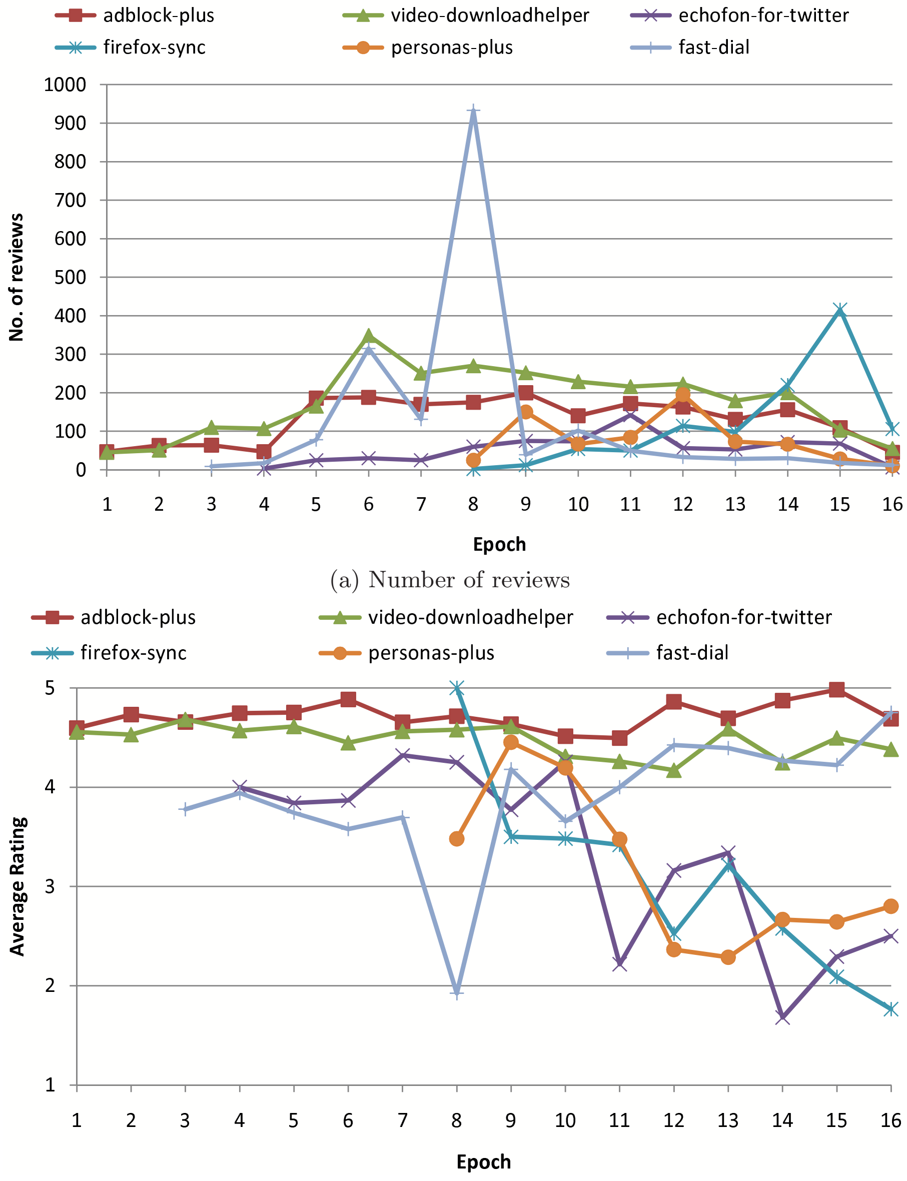
4.3.1 Dataset

In order to evaluate the dJST model, we crawled review documents between March 2007 and January 2011 from the Mozilla add-ons website[1]. These reviews relate to six different add-ons: *Adblock Plus*, *Video DownloadHelper*, *Firefox Sync*, *Echofon for Twitter*, *Fast Dial*, and *Personas Plus*. Compared to the publicly available datasets such as the movie review and multi-domain sentiment datasets, the Mozilla Add-on reviews have some distinct properties that are suitable for modelling sentiment and topic dynamics. One is that the number of reviews for each Add-on is generally large and the reviews are spread over an considerable time span; another is that for each review there is an associated user rating which can be used as a gold standard for evaluation. In the preprocessing, all the review documents were lower-cased and non-English characters were removed. We further removed the stop words from the documents based on a stop word list[2] and then performed Porter stemming [Porter, 2006]. Finally, the resulting dataset contains 9,114 documents, 11,652 unique words, and 158,562 word tokens in total.

After preprocessing, documents are sorted in an ascending order of their time stamps and divided into a sequence of quarterly epochs. Thus, from March 2007 to January 2011, there were 16 epochs. We plotted the total number of reviews for each add-on versus epoch number as shown in Figure 4.4a. 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 from Epoch 3 and 4, respectively. Reviews for *Firefox Sync*

[1] `https://addons.mozilla.org/`

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



(a) Number of reviews

(b) Average user rating

Figure 4.4: Dataset statistics. (a) number of reviews of each add-on over the epochs; (b) average user rating for each add-on over the epochs.

and *Personas Plus* only started to appear from Epoch 8. The review occurrences are strongly correlated with the release dates of the add-ons. We also noticed that there was 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 on a scale of 1 to 5 for expressing user's level of satisfaction on an add-on. 1-star indicates the lowest rating and 5-star indicates the highest. Figure 4.4b 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.

4.3.2 Evaluation Metrics

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

Predictive Perplexity Originally used in language modelling, perplexity measures a model's prediction ability on unseen data [Heinrich, 2005]. It is defined as the reciprocal geometric mean of the likelihood of a test corpus given a trained model's Markov chain state $\mathfrak{M}$. 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 $\tilde{\mathfrak{D}}_t = \{\tilde{\mathbf{w}}_d^t\}_{d=1}^{D^t}$ at epoch t based on the previously trained model $\mathfrak{M} = \{\mathbf{w}, \mathbf{z}, \mathbf{l}\}$ as

$$\text{Perplexity} = P(\tilde{\mathfrak{D}}_t|\mathfrak{M}) = \exp\{-\frac{\sum_{d=1}^{D^t} \log p(\tilde{\mathbf{w}}_d^t|\mathfrak{M})}{\sum_{d=1}^{D^t} \tilde{N}_d^t}\}, \tag{4.20}$$

where

$$P(\tilde{\mathbf{w}}_d^t|\mathfrak{M}) = \prod_{n=1}^{\tilde{N}_d^t} \sum_{l=1}^{L} \sum_{z=1}^{T} P(\tilde{w}_{d,n}|l,z)P(z|l)P(l), \tag{4.21}$$

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

$$P(\tilde{\mathbf{w}}_d^t|\mathfrak{M}) = \prod_{i=1}^{V} (\sum_{l=1}^{l} \sum_{z=1}^{T} \varphi_{l,z,i} \cdot \theta_{d,l,z} \cdot \pi_{d,l})^{\tilde{N}_{d,i}^t}, \tag{4.22}$$

where $\tilde{N}^t_{d,i}$ is the number of times word i has appeared in the dth document of the test set. Using Equation 4.20 and 4.22, the perplexity of unseen documents can then be calculated given a trained dJST model.

Sentiment Classification The document-level sentiment classification in dJST follows the criterion described in Section 3.4.4, i.e., based on the probability of sentiment label given a document $P(l|d)$. In terms of the gold standard sentiment labels, 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 of the existing approaches used balanced dataset for training (i.e., with the same number of positive and negative documents) to avoid the trained classifier bias towards a particular class, here we did not purposely make our dataset balanced and yet can achieve robust performance.

4.4 Experimental Results

In this section, we present the experimental results of the dJST model on the Mozilla add-on review dataset in terms of predictive perplexity, sentiment classification accuracy and topic evolution.

4.4.1 Performance vs. Number of Time Slices

As a dynamic model, dJST accounts for the historical context at previous epochs specified by a total number of S time slices. A larger time slice number indicates a longer history period modelled by dJST. In order to investigate the influence of the time slice number on the model performance, we vary $S \in \{1, 2, ..., 5\}$ and evaluate the model performance in both perplexity and sentiment classification accuracy. In our experiments, a model trained on the data at epoch $t-1$ is tested on the data of the next epoch t.

We have also compared the dJST model of incorporating the historical context in different ways, namely, the *sliding window*, *skip model*, and *multiscale model*. For all these models, the weight vector $\boldsymbol{\mu}^t$ for the evolutionary matrix is set either based on a decay function (denoted as -decay) or estimated directly from data using the fixed-point iteration method based on Equation 4.16 (denoted as -EM). We use three sentiment labels (i.e., positive, negative and neutral) and 15 topics under each sentiment label, which is

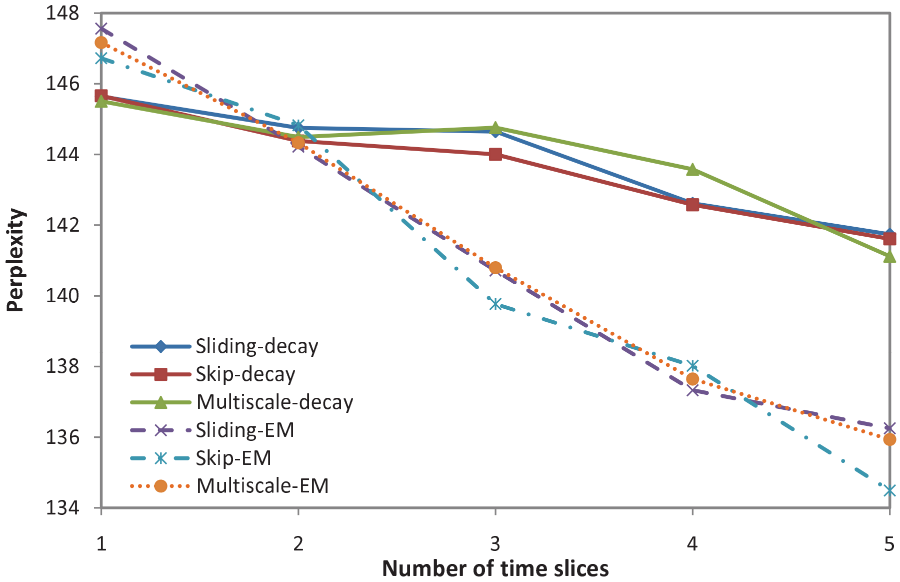
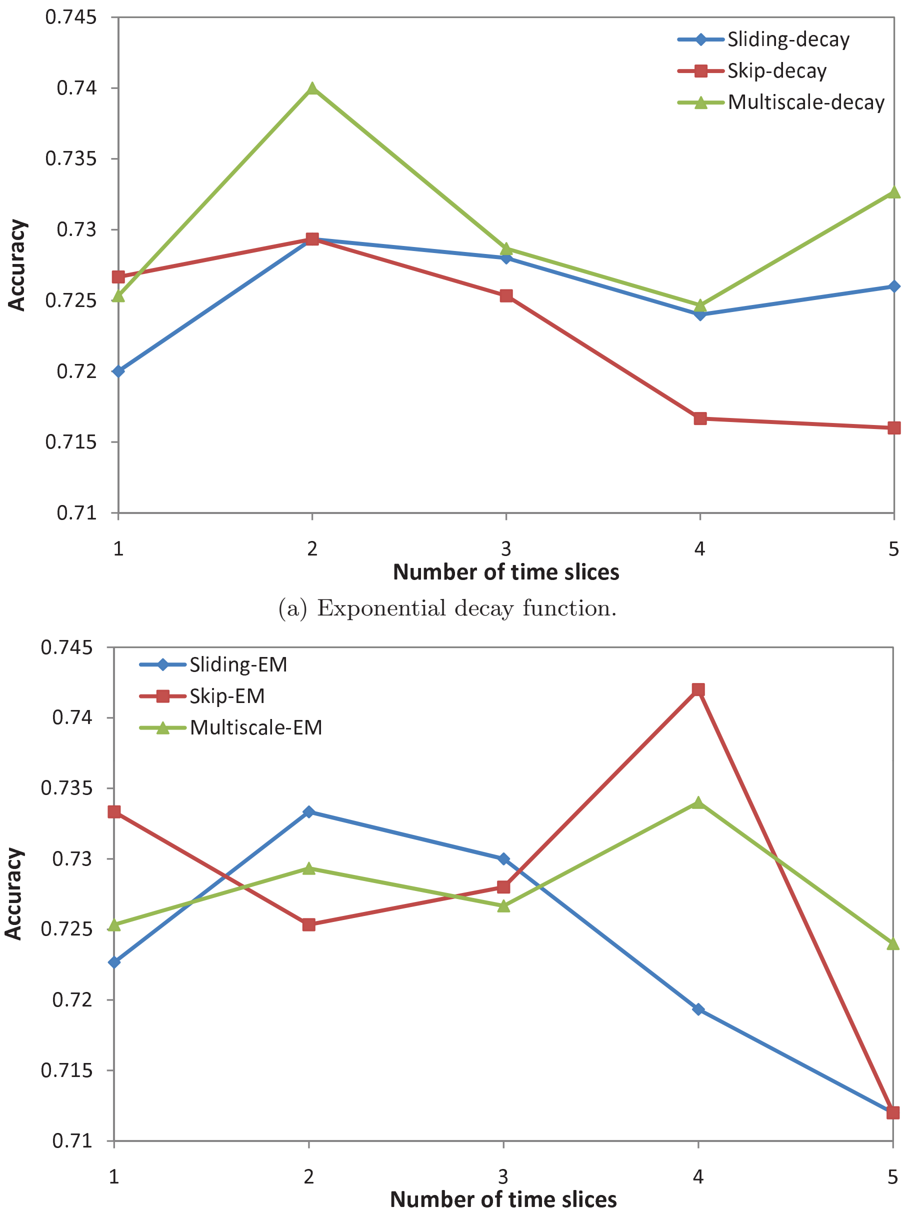


Figure 4.5: Perplexity vs. number of time slices.

equivalent to modelling 45 sentiment-topic clusters in total. Besides, the hyperparameter $\boldsymbol{\alpha}^t$ of all these models was optimized using the stochastic EM algorithm as described in Section 4.2.3.

Perplexity Results

Figure 4.5 shows the average perplexity over epochs against 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 above 4. Also, apart from the single time slice ($S = 1$) setting, models with their weight vector of the evolutionary matrix estimated from data using EM give lower perplexities than models with weights set with the decay function. This observation shows that by inferring the weights of the historical dependencies from data, dJST can better respond to the topic dynamics. When comparing the time slice models, it is observed that the *Skip model* slightly outperforms other models using a decay function, although it is not statistically significant, whereas all three models perform similarly when using EM for estimation.



(a) Exponential decay function.

(b) Fixed-point iteration method.

Figure 4.6: Accuracy vs. number of time slices.

Sentiment Classification Accuracy

Figure 4.6 shows the average sentiment classification accuracy against different number of time slices. It can be observed that accounting for historical context improves the sentiment classification accuracy for all three time slice models, where the best accuracy achieved is about 74% using both decay function and EM estimation. On the other hand, it is noted that modeling larger number of time slices does not necessary yield better accuracy. For instance, using the decay function, dJST peaks at $S = 2$ with the *MultiScale model*. In contrast, with the EM estimation, dJST attains the best result at $S = 4$ with the *Skip model*. One may also notice that the difference between accuracies using different number of time slices appear to be small. So in other context a different time slice setting may yield better results. We hypothesize that the difference of sentiment accuracy with different number of time slices could be more significant when tested on a larger dataset.

Although dJST achieves similar best performance in sentiment classification using the decay function and the EM estimation, its perplexity results with EM are much lower. Therefore, in all the subsequent experiments, we estimated the weight vector of the evolutionary matrix from data using EM unless otherwise specified.

4.4.2 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.

Perplexity for each epoch

Figure 4.7 shows the average perplexity of each epoch of different models. For dJST, we fix the number of time slices to 4 and set the number of topics to 15. For the non-dynamic models, we set 15 topics for both JST-one and JST-all, and 3 topics for LDA-one, which correspond to the positive, negative, and neutral sentiment labels. Word-polarity prior information was incorporated into LDA-one in a similar way to the dJST and JST models[1].

[1] 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.

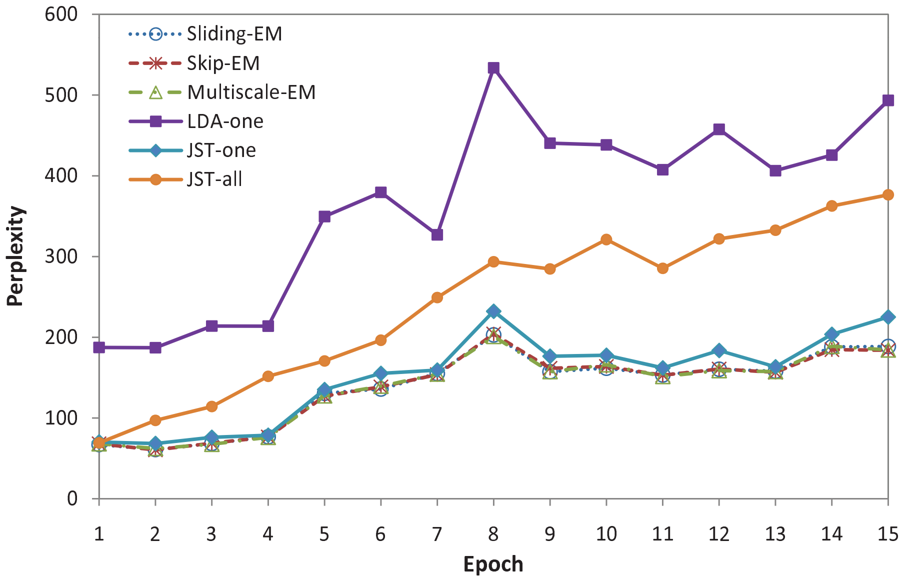


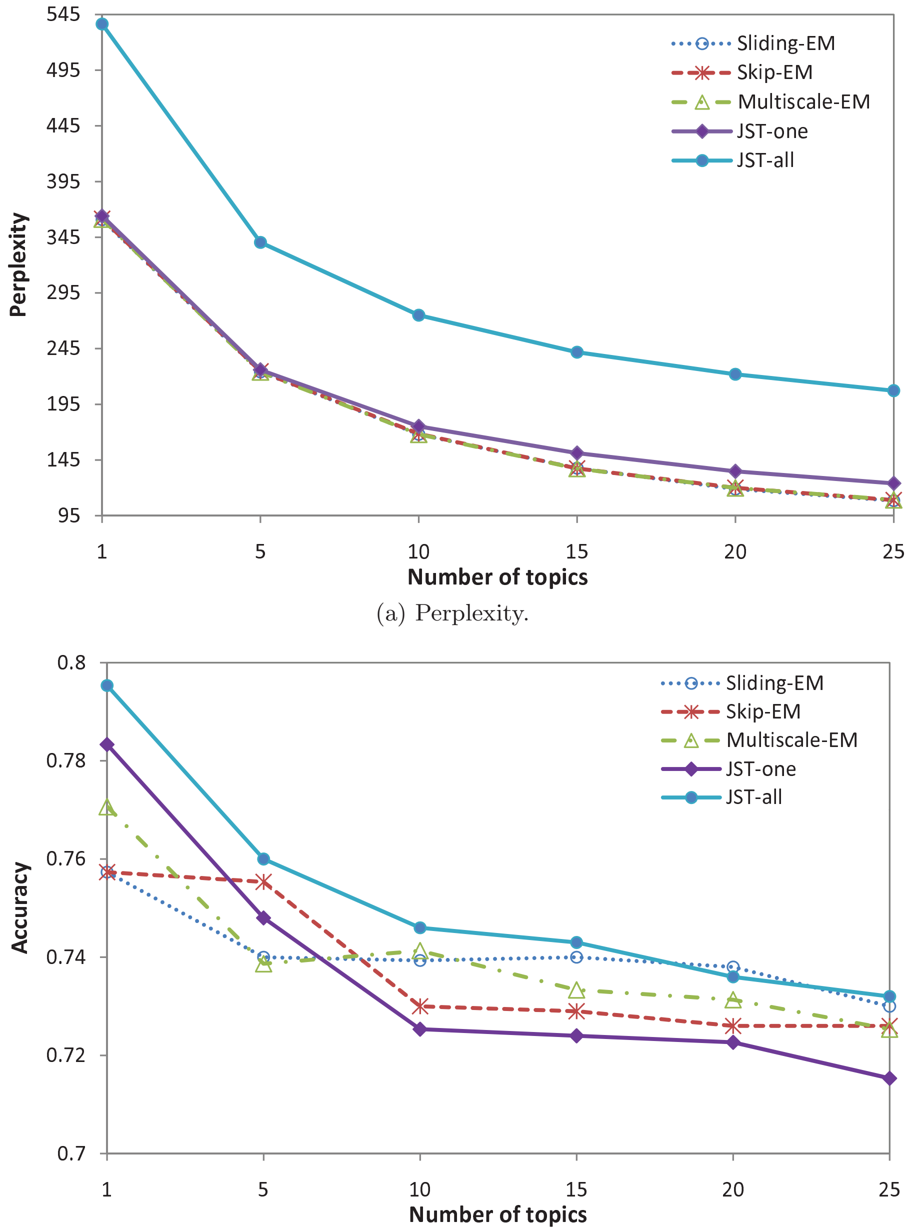
Figure 4.7: Perplexity for each epoch.

It is observed in Figure 4.7 that for all the models, perplexity generally increases as the epoch number increases. This observation is reasonable because different add-on reviews start appearing in the dataset from different epochs (cf. dataset statistics in Figure 4.4a) and they exhibit quite different distributions. Therefore a trained model is likely to lose predictive power when a test dataset contains reviews for a new type of add-on.

When comparing different models, 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 beyond epoch 8. Examining Figure 4.4 reveals that there is a very large volume of reviews about the add-on *Fast-dial* at epoch 8. This suggests that the distribution of the *Fast-dial* reviews significantly differs from other reviews, and thus modelling all the data from past epochs harms the JST-all model performance in predicting unseen data. In terms of the dJST model variants, they all outperform JST-one and JST-all and have quite similar perplexities.

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 4.8a shows the average per-word perplexity across the epochs. It can be observed that, for all the models, their perplexities decrease as the topic number increases and the perplexity values become saturated beyond $T = 20$.



(a) Perplexity.

(b) Sentiment classification accuracy.

Figure 4.8: Perplexity and sentiment classification accuracy versus number of topics.

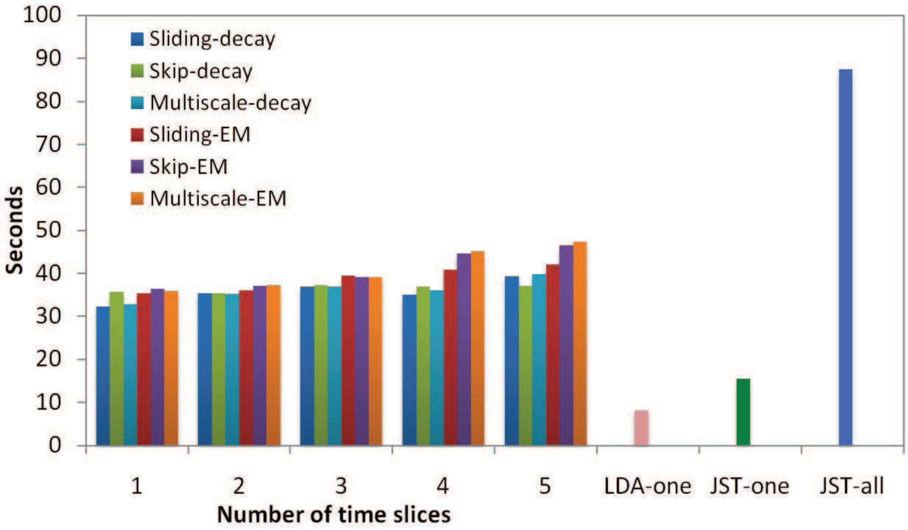


Figure 4.9: Average training time per epoch with different number of time slices.

Analogous to the observations in the previous experiment, all the variants of the dJST model have fairly similar perplexity values and they all outperform JST-one and JST-all, with JST-all having the highest perplexity. On the other hand, it is noted that the perplexity gap between the dJST models and JST-one increases continuously as the number of topics increases, which demonstrates the robustness of dJST to variations in the number of topics.

Figure 4.8b shows the average document-level sentiment classification accuracy over epochs with different number of topics. There are three main observations which can be made from the figures. First, the best classification accuracy was achieved with the single topic setting $T = 1$ for all the models. Second, increasing the topic number leads to a slight drop in accuracy, although accuracy becomes stabilised at $T = 10$ and beyond for all the models. Third, when comparing dJSTs with JST-one and JST-all, it was observed that dJST outperforms JST-one with each of the time slice models except at $T = 1$; whereas with sliding-EM and multiscale-EM, dJST attains similar sentiment classification accuracy as JST-all beyond $T = 10$.

We measured the significance of the perplexity and sentiment results with a paired t-test (critical P=0.01). Results show that while the dJST models significantly outperform JST-one and JST-all beyond $T = 10$ in perplexity, their performance in accuracy is statistically undistinguishable.

Computational Time

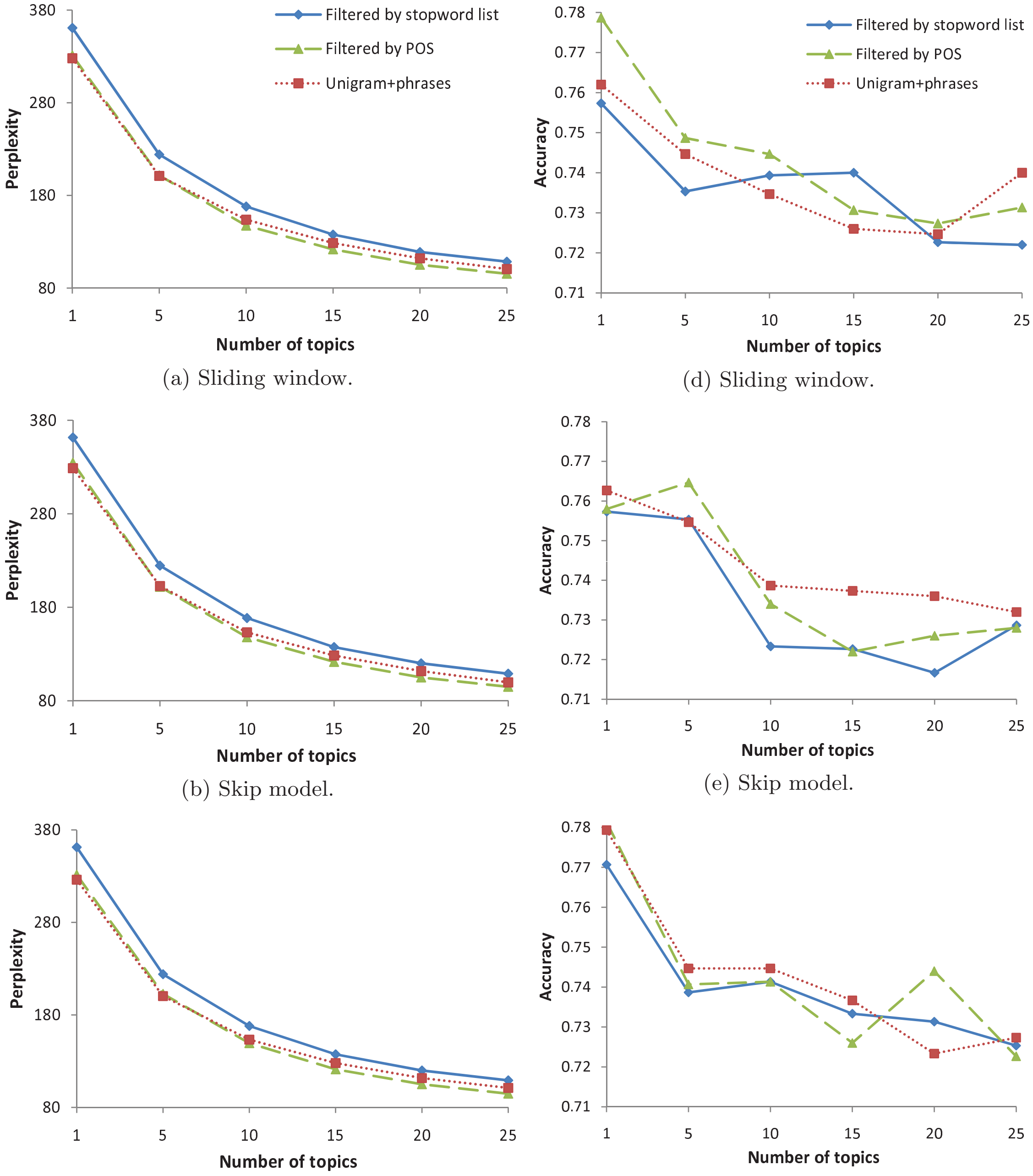
Figure 4.9 shows the average training time per epoch with the number of topics set to 15 using a computer with a dual core CPU of 2.8GHz and 2G of memory. Sliding, skip, and multiscale decay models have similar average training time across the number of time slices. For the dJST EM models, estimating the weight vector of the evolutionary matrix takes up more time, with its training time increasing linearly against the number of time slices. JST-one requires 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 longer time and space than all the other models as it needs to use all the previous data for training.

To conclude, the dJST model using the sliding-EM and multiscale-EM settings achieves much lower perplexity than JST-all while maintaining similar sentiment classification accuracies. Additionally, they avoid taking all the historical context into account and hence are computationally much more efficient than JST-all. On the other hand, the dJST models outperform JST-one in terms of both perplexity values and sentiment classification accuracies, which indicates the effectiveness of modelling dynamics.

4.4.3 Exploring Different Input Features

In the previous experiments, we pre-processed documents by removing stop words from a stop word list and used unigrams as input features for model learning. We 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 compared the performance of the dJST models using the original feature representation (*Filtered by stop word list*) to that based on the dataset preprocessed by two other strategies, i.e., 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 4.10, we set the number of time slices $S = 4$ and number of topics $T \in \{1, 5, 10, 15, 20, 25\}$.

The left panel of Figure 4.10 shows the average per-word perplexity over epochs for a different number of topics. It is observed that in general, increasing topic numbers results in lower perplexity values. In addition, dJSTs trained with features *Filtered by POS*



(a) Sliding window.

(d) Sliding window.

(b) Skip model.

(e) Skip model.

(c) Multiscale model.

(f) Multiscale model.

Figure 4.10: Performance vs. different input features. Left panel: perplexity; right panel: sentiment classification accuracy.

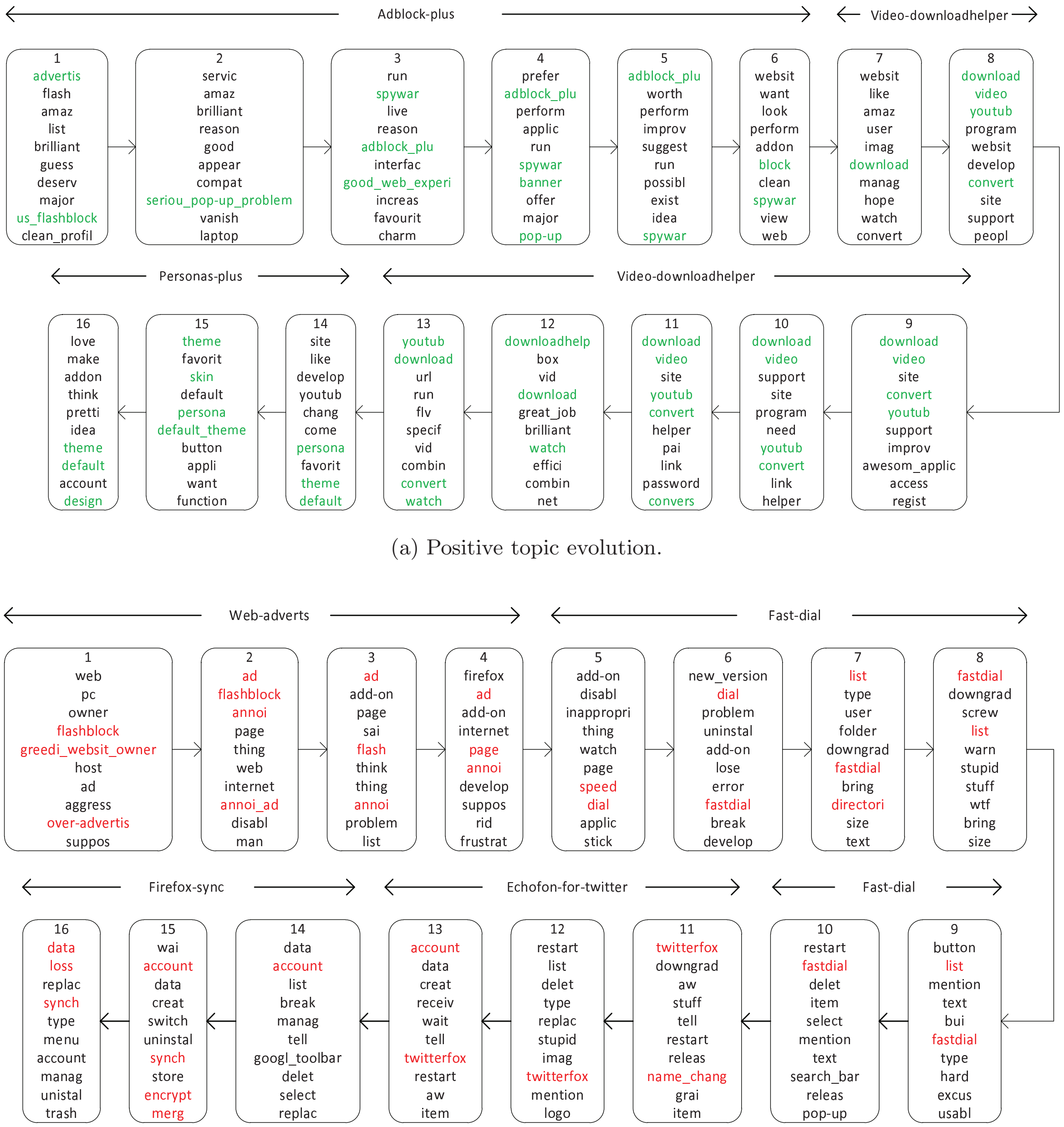
achieve slightly lower perplexity or than *Unigrams+phrases*, and they both give lower perplexities than the original feature representation (*Filtered by stop word list*) which is statistically significant based on paired t-test (critical P=0.01).

We also plot the average document-level sentiment classification accuracy over epochs with different number of topics as shown in the right panel of Figure 4.10. It can be observed that models trained with features *Filtered by POS* outperform *Filtered by stop word 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 the *multiscale model*.

4.4.4 Topic Evolution

One of the advantages of the dJST model over other non-dynamic topic models is that it can analyze the evolution of the sentiment-bearing topics over time. In order to demonstrate this feature offered by dJST, we list in Figure 4.11 one positive sentiment topic and one negative sentiment topic that evolve across the entire 16 epochs (i.e., from March 2007 to January 2011). These two topics are extracted by the dJST-multiscale model with the number of topics $T = 10$ and the number of time slices $S = 4$, based on the *Unigrams+phrases* features.

It was found in Figure 4.11 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 noticed that the negative phrase '*seriou_pop-up_problem*' appears in the positive topic at Epoch 2. A manual examination of the original review text reveals that this phrase actually appears 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 4.11a 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*. For the negative sentiment topics, more topic transitions are observed. Beginning with complaints about web adverts, the negative topic then transitions to negative comments about *Fast Dial*. It was noticed that at Epoch 8, there are a high volume of reviews about *Fast Dial* with an average rating of about 2 stars as illustrated in Figure 4.4. Hence, the negative sentiment topics about *Fast Dial* centred around Epoch 8. Subsequently,


Adblock-plus
Video-downloadhelper
1 advertis flash amaz list brilliant guess deserv major us_flashblock clean_profil
2 servic amaz brilliant reason good appear compat seriou_pop-up_problem vanish laptop
3 run spywar live reason adblock_plu interfac good_web_experi increas favourit charm
4 prefer adblock_plu perform applic run spywar banner offer major pop-up
5 adblock_plu worth perform improv suggest run possibl exist idea spywar
6 websit want look perform addon block clean spywar view web
7 websit like amaz user imag download manag hope watch convert
8 download video youtub program websit develop convert site support peopl
Personas-plus
Video-downloadhelper
16 love make addon think pretti idea theme default account design
15 theme favorit skin default persona default_theme button appli want function
14 site like develop youtub chang come persona favorit theme default
13 youtub download url run flv specif vid combin convert watch
12 downloadhelp box vid download great_job brilliant watch effici combin net
11 download video site youtub convert helper pai link password convers
10 download video support site program need youtub convert link helper
9 download video site convert youtub support improv awesom_applic access regist
Web-adverts
Fast-dial
1 web pc owner flashblock greedi_websit_owner host ad aggress over-advertis suppos
2 ad flashblock annoi page thing web internet annoi_ad disabl man
3 ad add-on page sai flash think thing annoi problem list
4 firefox ad add-on internet page annoi develop suppos rid frustrat
5 add-on disabl inappropri thing watch page speed dial applic stick
6 new_version dial problem uninstal add-on lose error fastdial break develop
7 list type user folder downgrad fastdial bring directori size text
8 fastdial downgrad screw list warn stupid stuff wtf bring size
Firefox-sync
Echofon-for-twitter
Fast-dial
16 data loss replac synch type menu account manag unistal trash
15 wai account data creat switch uninstal synch store encrypt merg
14 data account list break manag tell googl_toolbar delet select replac
13 account data creat receiv wait tell twitterfox restart aw item
12 restart list delet type replac stupid imag twitterfox mention logo
11 twitterfox downgrad aw stuff tell restart releas name_chang grai item
10 restart fastdial delet item select mention text search_bar releas pop-up
9 button list mention text bui fastdial type hard excus usabl

(a) Positive topic evolution.

(b) Negative topic evolution.

Figure 4.11: Example topics evolved over time. Topic labels shown above the boxed topics were derived manually from the coloured words and the number denotes epoch number. Topics in upper and lower panels are the positive and negative sentiment topics respectively.

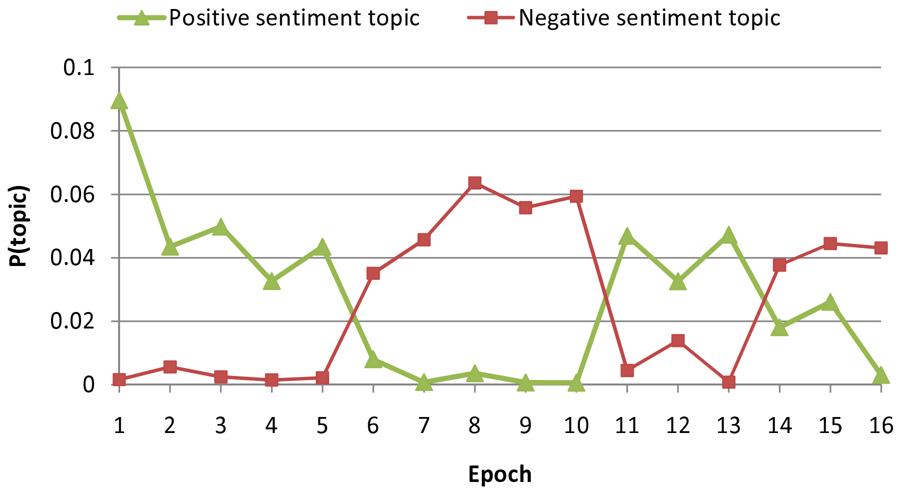


Figure 4.12: Occurrence probability of topics with time. Positive and negative sentiment topics correspond to the topics listed in the upper and lower panel of Figure 4.11 respectively.

the negative topic transitions to *Echofon for Twitter* at Epoch 11 and to *Firefox Sync* at Epoch 14.

Topic Prominence

The Mozilla add-on review dataset was collected over time, so the patterns presented in the early part of the collection may not be preserved in the latter. Such dynamics in the data can be reflected by the rise and fall in topic prominence.

We studied the prominence of the two topics presented in Figure 4.11 by plotting their topic occurrence probability against time, as shown in Figure 4.12. Here, the occurrence probability of a topic z occurred under a sentiment label l over the documents in each epoch t can be calculated as follows

$$P(z,l) = \frac{1}{|D_t|}\sum_{d=1}^{D_t} P(z|l,d)P(l|d) \tag{4.23}$$

$$= \frac{1}{|D_t|}\sum_{d=1}^{D_t} \theta^t_{d,l,z} \cdot \pi^t_{d,l}\,, \tag{4.24}$$

where D_t is the total number of documents in epoch t.

Figure 4.12 shows the prominence of the positive sentiment topics about *Adblock Plus* in the first five epochs. This observation is in line with the dataset statistics shown in Figure 4.4 that reviews on *Adblock Plus* take up a relative large portion in the first 5 epochs and the average user rating of this add-on is more than 4.5 stars over the entire epoch history. After Epoch 5, the negative sentiment topics become prominent as more low rating reviews start to emerge. For example, at Epoch 8 there are more than 900 reviews of *Fast dial* with an average rating of only 2 stars. A similar phenomenon can also be observed in Figure 4.12 that after Epoch 13, the negative sentiment topics become more prominent than the positive sentiment topics, which is again consistent with what we have observed in Figure 4.4a namely that there are an increasing number of reviews about *Firefox Sync* after Epoch 13 with an average user rating of only 2 stars. In conclusion, the above analysis demonstrates the effectiveness of the dJST model in analysing topic evolution of documents collected over time.

4.5 Discussion

In this chapter, we have presented the dynamic joint sentiment-topic (dJST) model, which models the 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 time slice settings to account for historical dependencies, namely, the *sliding window*, *skip model*, and *multiscale model*, as well as two strategies for estimating the weights of the dependencies using the decay function and EM estimation.

Experimental results on a real-world dataset demonstrate the effectiveness of dJST in terms of predictive likelihood and sentiment classification accuracy. It was found that dJST performs better with the EM estimation than using the decay function. Also, while dJST with the three different time slice settings gives similar perplexity values, both the *sliding window* and *multiscale model* generate slightly better sentiment classification results than the *skip model*.

The JST model presented in Chapter 3 and the dJST model discussed in this chapter are designed for sentiment classification at the document level. In the next chapter, we will introduce a new probabilistic topic model called the subjectivity detection LDA (subjLDA) model for finer-grained sentence-level subjectivity classification.

Chapter 5

SubjLDA: a Weakly Supervised Topic Model for Subjectivity Detection

5.1 Introduction

Subjectivity detection seeks to identify whether the given text expresses opinions (subjective) or reports facts (objective). Such a task of distinguishing subjective information from objective is useful for many natural language processing applications. For instance, it is often assumed in sentiment classification that the input documents are opinionated, and ideally contain subjective statements only [Yu and Hatzivassiloglou, 2003]; for question answering systems, extracting and presenting information of the appropriate type (i.e. opinions or facts) is imperative according to the specific question being asked [Wiebe and Riloff, 2005]. In this chapter, we present a hierarchical Bayesian model based on latent Dirichlet allocation (LDA) [Blei et al., 2003], called subjLDA for sentence-level subjectivity detection.

In contrast to most of the existing methods of subjectivity detection relying on either labelled corpora or linguistic pattern extraction for subjectivity classifier training [Murray and Carenini, 2009; Raaijmakers et al., 2008; Wilson and Raaijmakers, 2008], we view the problem as weakly-supervised generative model learning. The proposed subjLDA model can automatically identify whether a given sentence expresses opinions or states facts, and the only input to the model is a small set of domain independent subjectivity lexical clues. In the subjLDA model, the generative process involves three steps: (1) choose a subjectivity label for a sentence; here we define three possible subjectivity labels that a sentence expresses subjective opinions as being positive subjective or negative subjective, or states facts as being objective; (2) draw a sentiment label for each word in the sentence, where the sentiment label could be positive, negative, or neutral; and (3) draw the words in the sentence depending on the sentiment label.

We test the subjLDA model on the publicly available Multi-Perspective Question Answering (MPQA) dataset[1]. Two lists of domain independent subjectivity lexicons, namely the subjClue[2] [Wiebe and Riloff, 2005] and SentiWordNet lexicons[3] [Esuli and Sebastiani, 2006], were incorporated as prior knowledge for the subjLDA model learning. Detailed discussions of these two lexicons will be given in Section 5.3.2. Preliminary results show that the weakly-supervised subjLDA model is able to significantly outperform the baselines. Furthermore, it was found that while incorporating subjectivity clues bearing positive or negative polarity can achieve a significant performance gain, the prior lexical information from neutral words is less effective for improving the classification accuracy.

The rest of the chapter is organized as follows. Section 5.2 presents the subjLDA model and the inference algorithm. Experimental setup and results based on the MPQA dataset are described in Section 5.3 and 5.4, respectively. Finally, we summarize the work in Section 5.5

5.2 The SubjLDA Model

As shown in Figure 5.1, subjLDA is essentially a four-layer hierarchical Bayesian model. In order to generate a word $w_{d,m,t}$ which is the tth word token of sentence m within document d, one first chooses a subjectivity label $s_{d,m} \in \{1, ..., K\}$ for each sentence in document d from the per-document subjectivity proportion $\boldsymbol{\pi}_d$. In the implementation we set three possible subjectivity labels ($K = 3$), i.e., positive subjective, negative subjective and objective[4]. Following that, one chooses a sentiment label $l_{d,m,t} \in \{1, ..., S\}$ for each word in the sentence from the per-sentence sentiment proportion $\boldsymbol{\theta}_{s_{d,m}}$, where the sentiment labels could be positive, negative, or neutral ($S = 3$). Finally, one draws a word from the per-corpus word distribution $\boldsymbol{\varphi}_{l_{d,m,t}}$ conditioned on the corresponding sentiment label

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

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

[3] http://sentiwordnet.isti.cnr.it/

[4] We have conducted another set of experiments modelling two subjectivity labels only, i.e., either subjective or objective. It was found that subjLDA performed slightly better with three subjectivity labels than with binary labels and thus we do not report the binary label results here.

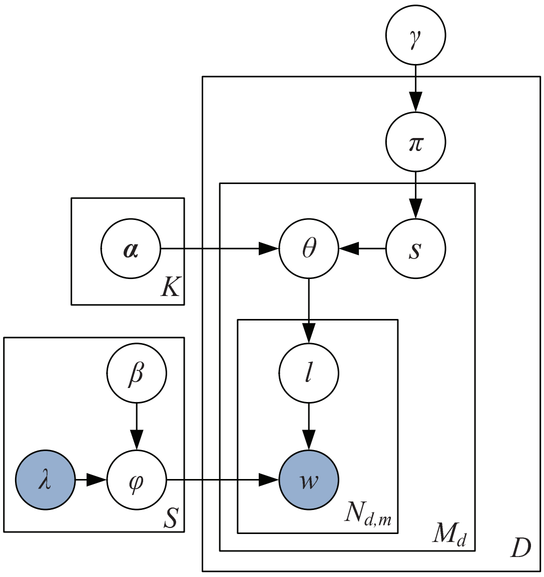


Figure 5.1: subjLDA model.

$l_{d,m,t}$. The classification of sentence subjectivity in subjLDA is then determined directly from the subjectivity label $s_{d,m}$ assigned to each sentence.

In order to encode the word-polarity prior information of the subjectivity lexicons into the subjLDA model, we modified the Dirichlet prior $\boldsymbol{\beta}$ on the per-corpus word distribution by a transformation matrix $\boldsymbol{\lambda}$ of size $S \times V$, where V is the total number of unique terms in the corpus. The procedures of encoding the prior knowledge exactly follow that described in Section 3.4.2. We summarize the notations used in this chapter in Table 5.1 and the formal definition of the subjLDA generative process corresponding to Figure 5.1 is as follows:

- For each sentiment label $l \in \{1, ..., S\}$
  - Draw $\boldsymbol{\varphi}_l \sim \mathrm{Dir}(\boldsymbol{\lambda}_l \cdot \boldsymbol{\beta}_l^T)$
- For each document $d \in \{1, ..., D\}$
  - Choose a distribution $\boldsymbol{\pi}_d \sim \mathrm{Dir}(\gamma)$
  - For each sentence $m \in \{1, ..., N_d\}$ in document d
    - Choose a subjectivity label $s_{d,m} \sim \mathrm{Mult}(\boldsymbol{\pi}_d)$,
    - Choose a distribution $\boldsymbol{\theta}_{d,m} \sim \mathrm{Dir}(\boldsymbol{\alpha}_{s_{d,m}})$
    - For each of the $N_{d,m}$ word position in sentence m of document d

- · Choose a sentiment label $l_{d,m,t} \sim \text{Mult}(\boldsymbol{\theta}_{s_{d,m}})$
- · Choose a word $w_{d,m,t} \sim \text{Mult}(\boldsymbol{\varphi}_{l_{d,m,t}})$

Table 5.1: Notations used for the subjLDA model.

Symbol	*Description*
D	number of documents in the collection.
K	number of subjectivity labels.
S	number of sentiment labels.
V	number of unique words
$\boldsymbol{\alpha}$	asymmetric Dirichlet priors on the sentiment mixing proportions, $\boldsymbol{\alpha} = \{\{\alpha_{k,j}\}_{j=1}^{S}\}_{k=1}^{K}$ ($K \times S$ matrix).
$\boldsymbol{\beta}$	asymmetric Dirichlet priors on the sentiment-word distribution, $\boldsymbol{\beta} = \{\{\beta_{j,r}\}_{r=1}^{V}\}_{j=1}^{S}\}$ ($S \times V$ matrix).
γ	symmetric Dirichlet priors on the subjectivity mixing proportions (scalar).
$\boldsymbol{\pi}_d$	parameter notation for the subjectivity mixing proportions for document d ($K-$ vector). For D documents, $\boldsymbol{\Pi} = \{\{\pi_{d,k}\}_{k=1}^{K}\}_{d=1}^{D}$ ($D \times K$ matrix).
$\boldsymbol{\theta}_{d,m}$	parameter notation for the sentiment mixing proportions for sentence m in document d ($S-$ vector). For D documents and N_d sentences in each document, $\boldsymbol{\Theta} = \{\{\{\theta_{d,m,j}\}_{j=1}^{S}\}_{m=1}^{N_d}\}_{d=1}^{D}$ ($D \times N_d \times S$ matrix).
$\boldsymbol{\varphi}_j$	parameter notation for the multinomial distribution over words for the sentiment label j. For S sentiment labels, $\boldsymbol{\Phi} = \{\{\varphi_{j,r}\}_{r=1}^{V}\}_{j=1}^{S}$ ($S \times V$ matrix).
$s_{d,m}$	subjectivity label for sentence m in document d.
$l_{d,m,t}$	sentiment label for word token t in sentence m of document d.
$w_{d,m,t}$	word t in sentence m of document d.
N_d	number of sentences in document d.
$N_{d,m}$	number of words in sentence m of document d.
$N_{d,k}$	number of sentences in document d are associated with subjectivity label k.
$N_{d,m,j}$	number of words in document d sentence m are associated sentiment label j.
$N_{j,r}$	number of times the word r is associated with sentiment label j.
N_j	number of times the words in the corpus are associated with sentiment label j.

5.2.1 Model Inference

The inference objective of the subjLDA model is to find the multinomial parameter sets $\{\boldsymbol{\Pi}, \boldsymbol{\Theta}, \boldsymbol{\Phi}\}$ that best explain the observed data. Like JST and dJST, exact inference is intractable in subjLDA, so we resort to the Gibbs sampling algorithm for approximate estimation of the model parameters. To obtain the Gibbs sampler for subjLDA, the full conditional distribution of the model hidden variables, here the subjectivity label s for a sentence and the sentiment label l for a word token must be found, which can be derived by evaluating the joint distribution of the model.

Given the model hyperparameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, the total probability of subjLDA for the words in the document collection $\mathbf{w}$ and the corresponding assignments for the subjectivity labels $\mathbf{s}$ and sentiment labels $\mathbf{l}$ can be factorized into three terms according to the model topology shown in Figure 5.1,

$$
\begin{aligned}
P(\mathbf{w},\mathbf{s},\mathbf{l}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= P(\mathbf{w}|\mathbf{l},\boldsymbol{\beta})P(\mathbf{l}|\mathbf{s},\boldsymbol{\alpha})P(\mathbf{s}|\boldsymbol{\gamma}) && (5.1)\\
&= \int P(\mathbf{w}|\mathbf{l},\boldsymbol{\beta})P(\mathbf{l}|\boldsymbol{\beta})\,d\boldsymbol{\Phi}\cdot\int P(\mathbf{l}|\boldsymbol{\Theta})P(\boldsymbol{\Theta}|\mathbf{s},\boldsymbol{\alpha})\,d\boldsymbol{\Theta}\cdot\\
&\quad \int P(\mathbf{s}|\boldsymbol{\Pi})P(\boldsymbol{\Pi}|\boldsymbol{\gamma})\,d\boldsymbol{\Pi}, && (5.2)
\end{aligned}
$$

where each term on the RHS contains only one model parameter which can be handled separately.

For the first term, by integrating out $\boldsymbol{\Pi}$ we obtain

$$
\begin{aligned}
P(\mathbf{s}|\boldsymbol{\gamma}) &= \int P(\mathbf{s}|\boldsymbol{\Pi})P(\boldsymbol{\Pi}|\boldsymbol{\gamma})\,d\boldsymbol{\Pi} && (5.3)\\
&= \int \prod_{d=1}^{D}\prod_{m=1}^{N_d} P(s_{d,m}|\boldsymbol{\pi}_d)P(\boldsymbol{\pi}_d|\boldsymbol{\gamma})\,d\boldsymbol{\pi}_d && (5.4)\\
&= \prod_{d=1}^{D}\int \prod_{m=1}^{N_d} P(s_{d,m}|\boldsymbol{\pi}_d)\frac{\Gamma(\sum_{k=1}^{K}\gamma_k)}{\prod_{k=1}^{K}\Gamma(\gamma_k)}\prod_{k=1}^{K}\pi_{d,k}^{\gamma_k-1}\,d\boldsymbol{\pi}_d && (5.5)\\
&= \prod_{d=1}^{D}\int \frac{\Gamma(\sum_{k=1}^{K}\gamma_k)}{\prod_{k=1}^{K}\Gamma(\gamma_k)}\prod_{k=1}^{K}\pi_{d,k}^{N_{d,k}+\gamma_k-1}\,d\boldsymbol{\pi}_d && (5.6)\\
&= \prod_{d=1}^{D}\frac{\Gamma(\sum_{k=1}^{K}\gamma_k)}{\prod_{k=1}^{K}\Gamma(\gamma_k)}\frac{\prod_{k=1}^{K}\Gamma(N_{d,k}+\gamma_k)}{\Gamma(N_d+\sum_{k=1}^{K}\gamma_k)}, && (5.7)
\end{aligned}
$$

where $N_{d,k}$ denotes the number of sentences in document d which are assigned to the subjectivity label k and $N_d = \sum_{k=1}^{K} N_{d,k}$.

For the second term, by integrating out $\boldsymbol{\Theta}$ we obtain

$$
\begin{aligned}
P(\mathbf{l}|\mathbf{s},\boldsymbol{\alpha}) &= \int P(\mathbf{l}|\boldsymbol{\Theta})P(\boldsymbol{\Theta}|\mathbf{s},\boldsymbol{\alpha})\,d\boldsymbol{\Theta} && (5.8)\\
&= \int \prod_{d=1}^{D}\prod_{m=1}^{N_d}\prod_{t=1}^{N_{d,m}} P(l_{d,m,t}|\boldsymbol{\theta}_{d,m})P(\boldsymbol{\theta}_{d,m}|\boldsymbol{\alpha},s_{d,m})\,d\boldsymbol{\theta}_{d,m} && (5.9)\\
&= \prod_{d=1}^{D}\prod_{m=1}^{N_d}\int \prod_{j=1}^{S}\theta_{d,m,j}^{N_{d,m,j}}\frac{\Gamma(\sum_{j=1}^{S}\alpha_{s_{d,m},j})}{\prod_{j=1}^{S}\Gamma(\alpha_{s_{d,m},j})}\prod_{j=1}^{S}\theta_{d,m,j}^{\alpha_{s_{d,m},j}-1}\,d\boldsymbol{\theta}_{d,m} && (5.10)\\
&= \prod_{d=1}^{D}\prod_{m=1}^{N_d}\frac{\Gamma(\sum_{j=1}^{S}\alpha_{s_{d,m},j})}{\prod_{j=1}^{S}\Gamma(\alpha_{s_{d,m},j})}\frac{\prod_{j=1}^{S}\Gamma(N_{d,m,j}+\alpha_{s_{d,m},j})}{\Gamma(N_{d,m}+\sum_{j=1}^{S}\alpha_{s_{d,m},j})}, && (5.11)
\end{aligned}
$$

where $N_{d,m,j}$ denotes the number of words in sentence m of document d which are assigned to the sentiment label j and $N_{d,m} = \sum_{j=1}^{S} N_{d,m,j}$.

For the third term, integrating out $\mathbf{\Phi}$ yields:

$$P(\mathbf{w}|\mathbf{l},\boldsymbol{\beta}) = \int P(\mathbf{w}|\mathbf{l},\boldsymbol{\beta})P(\mathbf{l}|\boldsymbol{\beta})\,d\mathbf{\Phi} \tag{5.12}$$

$$= \int \prod_{j=1}^{S}\prod_{d=1}^{D}\prod_{m=1}^{N_d}\prod_{t=1}^{N_{d,m}} P(w_{d,m,t}|\boldsymbol{\varphi}_j)P(\boldsymbol{\varphi}_j|\boldsymbol{\beta})\,d\boldsymbol{\varphi}_j \tag{5.13}$$

$$= \prod_{j=1}^{S}\int \prod_{r=1}^{V}\varphi_{j,r}^{N_{j,r}} \frac{\Gamma(\sum_{r=1}^{V}\beta_{j,r})}{\prod_{r=1}^{V}\Gamma(\beta_{j,r})}\prod_{r=1}^{V}\varphi_{j,r}^{\beta_{j,r}-1}\,d\boldsymbol{\varphi}_j \tag{5.14}$$

$$= \prod_{j=1}^{S}\frac{\Gamma(\sum_{r=1}^{V}\beta_{j,r})}{\prod_{r=1}^{V}\Gamma(\beta_{j,r})}\frac{\prod_{r=1}^{V}\Gamma(N_{j,r}+\beta_{j,r})}{\Gamma(N_j+\sum_{r=1}^{V}\beta_{j,r})}, \tag{5.15}$$

where $N_{j,r}$ denotes the number of times the word r is assigned to the sentiment label j and $N_j = \sum_{r=1}^{V} N_{j,r}$.

Substituting Equations 5.7, 5.11 and 5.15 into 5.2 yields the analytical expression for 5.1 as follows,

$$P(\mathbf{w},\mathbf{l},\mathbf{s}|\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) = \prod_{d=1}^{D}\frac{\Gamma(\sum_{k=1}^{K}\gamma_k)}{\prod_{k=1}^{K}\Gamma(\gamma_k)}\frac{\prod_{k=1}^{K}\Gamma(N_{d,k}+\gamma_k)}{\Gamma(N_d+\sum_{k=1}^{K}\gamma_k)}\cdot$$
$$\prod_{d=1}^{D}\prod_{m=1}^{N_d}\frac{\Gamma(\sum_{j=1}^{S}\alpha_{s_{d,m},j})}{\prod_{j=1}^{S}\Gamma(\alpha_{s_{d,m},j})}\frac{\prod_{j=1}^{S}\Gamma(N_{d,m,j}+\alpha_{s_{d,m},j})}{\Gamma(N_{d,m}+\sum_{j=1}^{S}\alpha_{s_{d,m},j})}\cdot$$
$$\prod_{j=1}^{S}\frac{\Gamma(\sum_{r=1}^{V}\beta_{j,r})}{\prod_{r=1}^{V}\Gamma(\beta_{j,r})}\frac{\prod_{r=1}^{V}\Gamma(N_{j,r}+\beta_{j,r})}{\Gamma(N_j+\sum_{r=1}^{V}\beta_{j,r})}. \tag{5.16}$$

Having obtained the joint distribution, the task is then to derive the full conditional distributions for the model hidden variables, which are the subjectivity label s for each sentence and the sentiment label l for each word token.

Full Conditional Distribution for the Hidden Variable s Let $x = (d, m)$ denote the index for sentence m of document d, and the subscript $\neg x$ indicate the corresponding counts of the sentence with index x being excluded from a quantity. Omitting the hyperparameters and using the joint distribution in Equation 5.16, the full conditional posterior from which the Gibbs sampler draws the subjectivity label s_x for a sentence is:

$$P(s_x = k|\mathbf{s}_{\neg x},\mathbf{l},\mathbf{w}) = \frac{P(\mathbf{w},\mathbf{s},\mathbf{l})}{P(\mathbf{w},\mathbf{s}_{\neg x},\mathbf{l})} \tag{5.17}$$

$$\propto \frac{N_{d,k}^{\neg x}+\gamma_k}{N_d^{\neg x}+\sum_{k=1}^{K}\gamma_k}\cdot\frac{\prod_{j=1}^{S}\prod_{b=0}^{N_{d,m,j}-1}(b+\alpha_{s_x,j})}{\prod_{b=0}^{N_{d,m}-1}(b+\sum_{j=1}^{S}\alpha_{s_x,j})}. \tag{5.18}$$

Full Conditional Distribution for the Hidden Variable l Let the index $y = (d, m, t)$ denote the tth word in sentence m of document d and the subscript $\neg y$ indicate excluding the counts of the word token with index y from a quantity. Omitting the hyperparameters and applying the joint distribution of subjLDA yields the full conditional posterior for l_y,

$$P(l_y = j|\mathbf{s}, \mathbf{l}_{\neg y}, \mathbf{w}) = \frac{P(\mathbf{w}, \mathbf{s}, \mathbf{l})}{P(\mathbf{w}, \mathbf{s}, \mathbf{l}_{\neg y})} \tag{5.19}$$

$$\propto \frac{N_{d,m,j}^{\neg y} + \alpha_{s_{d,m},j}}{N_{d,m}^{\neg y} + \sum_{j=1}^{S} \alpha_{s_{d,m},j}} \cdot \frac{N_{j,r}^{\neg y} + \beta_r}{N_j^{\neg y} + \sum_{r=1}^{V} \beta_r} \tag{5.20}$$

Using Equations 5.18 and 5.20, the Gibbs sampling procedures can be run by iteratively sampling the hidden variables $\boldsymbol{s}$ and $\boldsymbol{l}$. Samples obtained from the Markov chain are then used to approximate the subjLDA model parameters as follows.

The approximated per-document subjectivity proportion is

$$\pi_{d,k} = \frac{N_{d,k} + \gamma_k}{N_d + \sum_{k=1}^{K} \gamma_k}. \tag{5.21}$$

The approximated per-sentence sentiment proportion is

$$\theta_{d,m,j} = \frac{N_{d,m,j} + \alpha_{s_{d,m},j}}{N_{d,m} + \sum_{j=1}^{S} \alpha_{s_{d,m},j}}. \tag{5.22}$$

Finally, the approximated per-corpus sentiment-word distribution is

$$\varphi_{j,r} = \frac{N_{j,r} + \beta_{j,r}}{N_j + \sum_{r=1}^{V} \beta_{j,r}}. \tag{5.23}$$

The Gibbs sampling procedure for subjDLA is summarized in Algorithm 3.

5.3 Experimental Setup

5.3.1 Dataset

We tested the subjLDA model on the MPQA dataset[1] version 1.2, which is derived from 187 different foreign and U.S. news sources. The whole corpus consists of 535 documents with a total number of 6,111 subjective and 5,001 objective sentences which are manually annotated. Although the dataset provides very fine-grained expression level subjectivity annotation, we only used the sentence-level subjectivity label as gold standard for evaluation.

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

Algorithm 3 Gibbs sampling procedure for subjLDA.

```
Input: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, Corpus
Output: sentiment label assignments for all the words and subjectivity label assignments
    for all the sentences in the corpus
Initialize all count variables $N_{d,k}$, $N_d$, $N_{d,m,j}$, $N_{d,m}$, $N_{j,r}$, $N_j$.
Randomize the order of documents in the corpus, the order of sentences in each document,
    and the order of the words in each sentence.
for $i = 1$ to $max$ Gibbs sampling iterations do
  for document $d \in \{1, ..., D\}$ do
    for sentence $m \in \{1, ..., N_d\}$ do
      Exclude sentence $m$ and its assigned subjectivity label $k$ from the variables
          $N_{d,k}$, $N_d$
      Sample a new subjectivity label $\tilde{s}_{d,m}$ for sentence $m$ using Equation 5.18
      Update variables $N_{d,k}$, $N_d$ using the new subjectivity label $\tilde{s}_{d,m}$
      for word token $t \in \{1, ..., N_{d,m}\}$ do
        Exclude word token $t$ and its assigned sentiment label $l$ from the variables
            $N_{d,m,j}$, $N_{d,m}$, $N_{j,r}$, $N_j$
        Sample a new sentiment label $\tilde{l}_{d,m,t}$ using Equation 5.20
        Update variables $N_{d,m,j}$, $N_{d,m}$, $N_{j,r}$, $N_j$ using the new sentiment label $\tilde{l}_{d,m,t}$
      end for
    end for
  end for
  for every 40 Gibbs sampling iterations do
    Update hyperparameter $\boldsymbol{\alpha}$ with maximum-likelihood estimation
  end for
  for every 200 Gibbs sampling iterations do
    Update the matrix $\boldsymbol{\Pi}$, $\boldsymbol{\Theta}$, and $\boldsymbol{\Phi}$ with the new sampling results
  end for
end for
```

A two-stage preprocessing was performed on the MPQA dataset by first removing stop words and non-word characters, followed by standard Porter stemming for reducing vocabulary size and minimizing data sparsity problems. After preprocessing, the MPQA dataset contains 131,220 words with 10,511 distinct terms (cf. the original dataset with 264,808 words and a vocabulary size of 31,201 without any preprocessing).

5.3.2 Lexical Prior Knowledge

We explored incorporating two subjectivity lexicons as prior knowledge for subjLDA model learning, namely, the subjClue[1] and SentiWordNet[2] lexicons. We point out that the subjClue lexicon is not related to the MPQA dataset as it was collected from a number

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

[2] `http://sentiwordnet.isti.cnr.it/`

of sources, where some were culled from manually developed resources and others were identified automatically using both annotated and unannotated data [Wiebe and Riloff, 2005]. We only extract the lexical clues that are considered strongly subjective, with the weakly subjective clues being discarded. The rationale behind the filtering is that while a strongly subjective clue is seldom used without a subjective meaning, weakly subjective clues are ambiguous, often having both subjective and objective uses. After stemming, removing the duplicated lexical terms and retaining those that have appeared in the MPQA corpus, we finally obtained a lexicon subset of 477 positive and 917 negative words.

SentiWordNet provides a wide coverage of lexical terms by tagging all the synsets of WordNet with three sentiment labels, i.e., positive, negative and neutral. In our experiment, we only use the neutral words from SentiWordNet for investigating how neutral words would affect the subjLDA model performance. After the same preprocessing as performed on the subjClue lexicon, a total of 193,871 neutral words were extracted. Further mapping the extracted neutral words with the corpus results in 6,457 neutral words.

In practice, it is quite intuitive that one classifies a sentence as subjective if it contains one or more strongly subjective clues [Riloff and Wiebe, 2003]. However, the criterion for classifying objective sentences could be rather different, because a sentence is likely to be objective if there are no strongly subjective clues. In order to encode this knowledge into the subjLDA model learning, during the model initialization step, we initialized sentence subjectivity label s based on the aforementioned criterion using the subjectivity lexicons as input. Specifically, if a sentence contains subjectivity clues, it will be assigned a subjective label, and objective label otherwise.

5.3.3 Hyperparameter Setting

In the subjLDA model implementation, we adopted similar strategies as described in Section 3.4.3 for hyperparameter setting. We set the asymmetric prior $\boldsymbol{\beta} = 0.01$ in the initialization [Steyvers and Griffiths, 2007], the symmetric prior $\{\gamma_k\}_{k=1}^{K} = \gamma = (0.05 \times L)/K$, where L is the average document length, and the value of 0.05 on average allocates 5% of probability mass for mixing. The asymmetric prior $\boldsymbol{\alpha}$ is learned directly from data using maximum-likelihood estimation [Minka, 2003] and updated every 40 iterations during the Gibbs sampling procedure.

5.4 Experimental Results

In this section, we first present the experimental results of sentence-level subjectivity classification on the MPQA dataset, and then evaluate the impact of the prior information on the classification performance by varying the proportion of subjectivity clues being incorporated. All the results reported here are averaged over 5 runs with 800 Gibbs sampling iterations, with 200 Gibbs sampling iterations discarded in-between each saved sample.

5.4.1 Subjectivity Classification Results

We compare subjLDA with the baseline and the LDA model [Blei et al., 2003] on the sentence-level subjectivity detection task. The baseline is calculated by counting the overlap of the prior subjectivity clues with the dataset. A sentence is then classified as subjective if it contains one or more strongly subjective words; if there is no matching, the sentence will be classified as objective. The improvement over this baseline will reflect how much subjLDA can learn from data.

For the LDA model, we set the number of topics $T = 3$ to model a mixture of three sentiment topics, i.e., positive, negative and neutral. For fair comparison, we also incorporated the prior knowledge of the subjectivity lexicons into LDA in the same way as subjLDA. Thus the LDA model here can be considered as a weakly-supervised version. Moreover, we tested LDA in two different modes. The LDA model in the document mode models the document collection in a normal way, i.e., each document contains multiple sentences, whereas in the sentence mode, each sentence was treated as an individual document. The sentence subjectivity label for the LDA models is determined as follows.

(a) LDA in document mode Given the mth sentence in the dth document $C_{d,m}$, the probability of observing sentiment label l for the sentence is

$$
\begin{aligned}
P(l|C_{d,m}) &\propto P(C_{d,m}|l)P(l|d) && (5.24)\\
&= \prod_{w_t \in C_{d,m}} P(w_t|l_{w_t})P(l_{w_t}|d) && (5.25)\\
&= \prod_{w_t \in C_{d,m}} \varphi_{l_{w_t},w_t} \cdot \theta_{d,l_{w_t}}, && (5.26)
\end{aligned}
$$

where w_t is the word token from sentence $C_{d,m}$, $\varphi_{l_{w_t},w_t}$ is the component of the multinomial distribution for sentiment label l_{w_t} and term w_t. $\theta_{d,l_{w_t}}$ is the component for the

Table 5.2: Subjectivity classification results. (Boldface indicates the best results.)

Model	Objective (%)			Subjective (%)			Overall (%)
	Recall	Precision	F-measure	Recall	Precision	F-measure	Accuracy
Baseline	46.5	74.1	57.1	76.7	63.7	69.6	63.1
subjLDA	59.7	**71.6**	**65.1**	**80.9**	**71.0**	**75.6**	**71.2**
LDA (Sent.)	**60.5**	65.7	63.0	74.2	69.7	72.0	68.1
LDA (Doc.)	51.4	68.7	58.8	80.6	67.0	73.2	67.6
Wiebe [2005]	77.6	68.4	72.7	70.6	79.4	74.7	73.8

multinomial distribution of document d and sentiment label l_{w_t}. We say that sentence $C_{d,m}$ is classified as an objective sentence if its probability of neutral label given sentence $P(l = neu.|C_{d,m})$ is greater than both $P(l = pos.|C_{d,m})$ and $P(l = neg.|C_{d,m})$. Otherwise, the sentence is classified as subjective.

(b) LDA in sentence mode Under the sentence mode, the probability of observing sentiment label l given the sentence $C_{d,m}$ can be obtained directly from the per-sentence sentiment proportion $\boldsymbol{\theta}_{d,m}$. The sentence subjectivity is then determined using the classification metric identical to the document mode.

As can be seen from Table 5.2, a significant performance gain was observed for both subjLDA and LDA over the baseline. Particularly, more than 8% gain was observed for subjLDA, giving the best overall accuracy of 71.2% which is 3.1% and 3.6% higher than LDA(Sent.) and LDA(Doc.), respectively. In addition, except for objective recall, subjLDA outperforms LDA in both the sentence and document modes for all the other evaluation metrics. On the other hand, it was observed that while LDA(Doc.) can achieve a comparable subjective F-measure to LDA(Sent.), its objective F-measure is nearly 5% lower, resulting in worse overall performance. This is probably due to the fact that by treating each individual sentence as a document, LDA(Sent.) can avoid inferencing global sentiment topics and thus capture salient local sentiment topics. We measured the overall accuracy significance with a paired t-test (critical P=0.01). Results show that the improvements of subjLDA over both LDA(sent.) and LDA(doc.) are highly statistically significant. Thus, we conclude that subjLDA is superior to LDA in the subjectivity detection task.

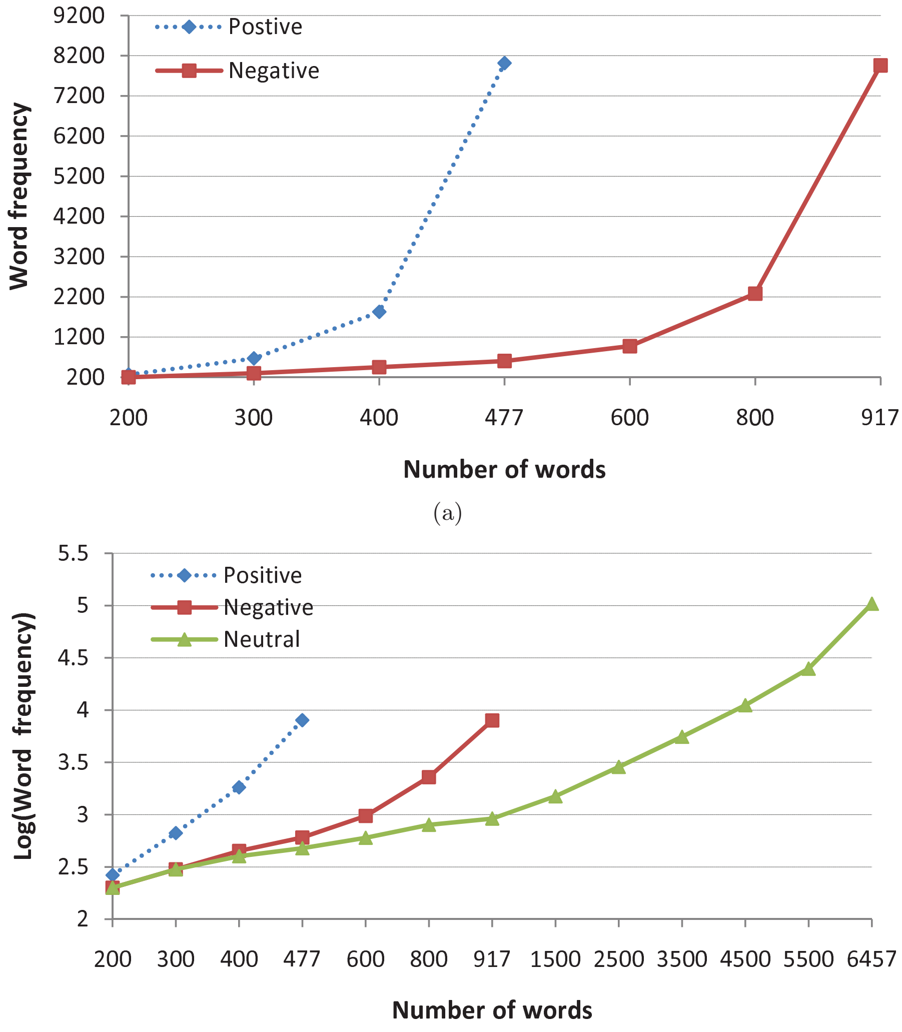
When compared to the previous proposed bootstrapping approach [Wiebe and Riloff, 2005], subjLDA is about 2% lower in terms of overall accuracy. However, it should be noted that, the approach of Wiebe and Riloff [2005] used a much larger training set for self-training which consists of more than 100,000 sentences. Moreover, apart from subjectivity clues, they also used additional features such as subjective/obejctive pattern and POS for the Naive Bayes sentence classifier training. In contrast, the proposed subjLDA model is relatively simple with only a small set of subjectivity clues being incorporated as prior knowledge.

5.4.2 Performance vs. Incorporating Different Priors

While subjectivity clues bearing positive or negative sentiment are commonly used in lexical approaches to sentiment classification, the impact of incorporating neutral words for subjectivity detection remains relatively unexplored. In this experiment, we investigated the model performance on subjectivity detection by incorporating additional prior knowledge from the neutral words. We started by first considering the positive and negative words only and gradually increased the number of positive and negative words starting with the lowest frequency ones. After all the positive and negative words have been incorporated, we then gradually added additional neutral words into the model also from the lowest frequency to the highest. Figure 5.2 shows the lexicon statistics of the positive, negative and neutral words being incorporated as prior knowledge, where the value on the x-axis represents the number of words sorted by word frequency and the corresponding y-axis value indicates the total number of times those words appear in the corpus. For instance, the 400 least frequent positive words appear a total of 1,826 times in the corpus, as shown in Figure 5.2a.

Figure 5.3 depicts the subjectivity classification results of subjLDA and LDA by varying the proportion of lexical terms being incorporated. It is quite obvious from the overall accuracy shown in the figure that both subjLDA and LDA benefit from incorporating the prior knowledge of the subjective words, and in general, the more lexical items the better the results. Without using any neutral words, all three models achieved the best results when all the subjective words were incorporated. It was noted that subjLDA performed similar to LDA when only a small number of low frequency subjective words were used. However, with more higher frequency subjective words being incorporated, subjLDA shows stronger performance boosting over LDA and gives the best accuracy of 70.2% when all the subjective words were incorporated, being 3.4% and 5.8% better than LDA(Sent.) and LDA(Doc.), respectively, as indicated by the vertical dashed line in the figure.

On the other hand, adding neutral words is also beneficial, where about 2% performance gain was observed for all the three models in addition to the best results using subjective words only. Analysing the objective recall and precision shown in Figure 5.3b



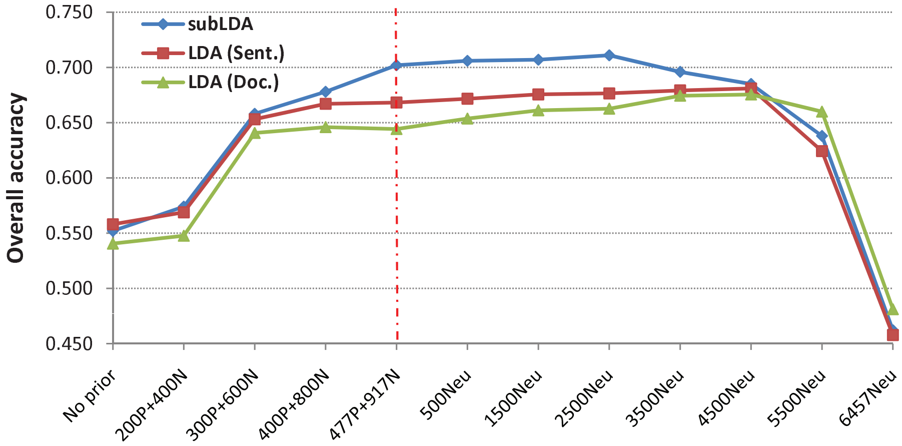
(a)

(b)

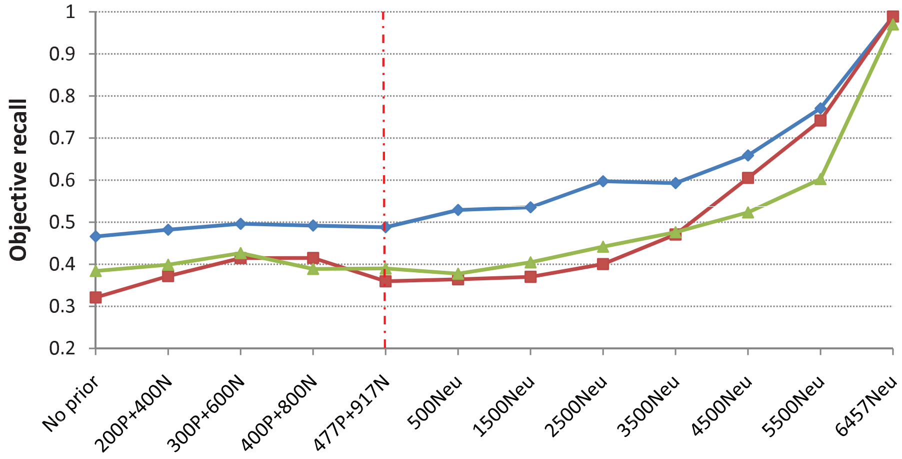
Figure 5.2: Sentiment lexicon statistics.

and 5.3c reveals that, while incorporating the 4,500 least frequent neutral words considerably increases the objective recall, the decrease in the objective precision is relatively small which eventually leads to the overall performance improvement of all the three models.

However, compared to the subjective words, the classification improvement by incorporating additional neutral words is less significant. This is probably due to the fact that while the presence of the subjectivity clues bearing positive or negative sentiment conveys clear subjective meanings, neutral words are relatively vague which could bear objective or subjective sense under different contexts. Furthermore, all three models experience a significant performance drop after the point of (4500Neu). Examining Figure 5.2 reveals that, while the 4,500 least frequent neutral words appear 11,142 times in the corpus, the 1,957 most frequent words (i.e., from 4500 to 6457) appear 93,036 times, nearly 10 times

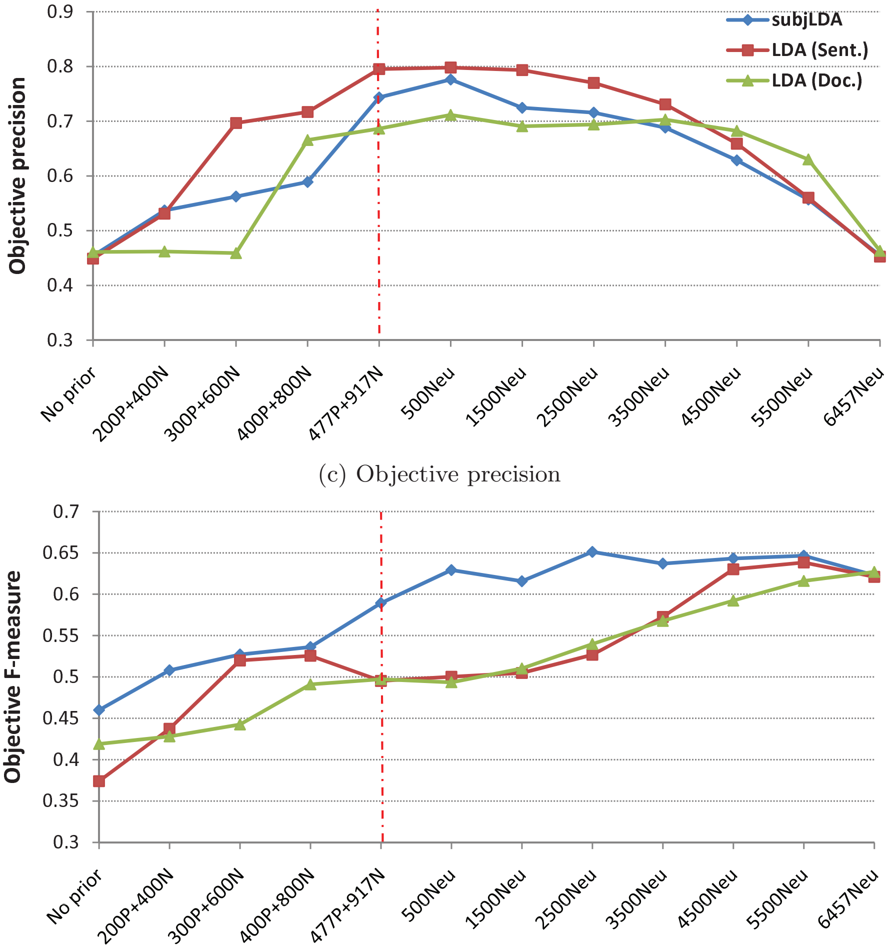


(a) Overall accuracy.



(b) Objective recall.

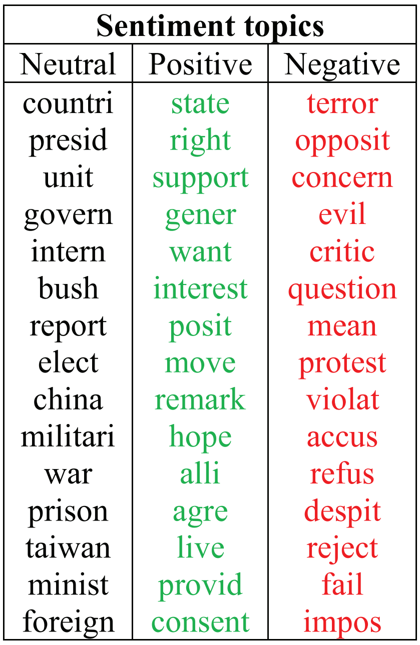
Figure 5.3: Subjectivity classification performance vs. different prior information by gradually adding the subjective and neutral words. The vertical dashed line denotes the point where all the positive and negative words have been incorporated into the model; 200P+400N denotes adding the least frequent 200 positive and 400 negative words. For example, 500Neu denotes adding the least frequent 500 neutral words in addition to all the positive and negative words.



(c) Objective precision

(d) Objective F-measure

Figure 5.3: Subjectivity classification performance vs. different prior information by gradually adding the subjective and neutral words. The vertical dashed line denotes the point where all the positive and negative words have been incorporated into the model; 200P+400N denotes adding the least frequent 200 positive and 400 negative words. For example, 500Neu denotes adding the least frequent 500 neutral words in addition to all the positive and negative words.



Sentiment topics		
Neutral	Positive	Negative
countri	state	terror
presid	right	opposit
unit	support	concern
govern	gener	evil
intern	want	critic
bush	interest	question
report	posit	mean
elect	move	protest
china	remark	violat
militari	hope	accus
war	alli	refus
prison	agre	despit
taiwan	live	reject
minist	provid	fail
foreign	consent	impos

Figure 5.4: Sentiment topics extracted by subjLDA.

as much as the former. Thus, the high frequency neutral words become dominant in the model and result in severe classification bias towards the objective class. Therefore, appropriate filtering of neutral words is necessary in order to avoid introducing bias into model learning.

5.4.3 Sentiment Topics

In subjLDA, we model three topics in the per-corpus word distribution, each of which corresponds to the neutral, positive and negative sentiment label. Figure 5.4 shows the top 15 topic words of three sentiment topics extracted from the MPQA dataset by subjLDA. It can be easily observed that while the positive and negative sentiment topics contain clear sentiment bearing words such as *support*, *interest*, *terror*, *opposit* etc., the neutral topic contains mostly theme words with no sentiment, which illustrates the effectiveness of subjLDA in extracting sentiment bearing topics from text.

5.5 Discussions

This chapter has presented the subjectivity detection LDA Model (subjLDA) for sentence-level subjectivity classification. In contrast to most of the existing approaches to subjectivity detection requiring labelled corpora or linguistic pattern extraction for classifier training, we view the problem as weakly-supervised generative model learning where the only supervised information used in the model is a small amount of domain independent subjectivity and neutral words.

The subjLDA model has been evaluated on the MPQA dataset. Preliminary results show that except slightly lower in objective recall, subjLDA outperformed LDA over all other evaluation metrics and is comparable to the previously proposed bootstrapping approach using a much larger dataset for training. It was also found that while incorporating more subjective words can generally yield better results, the performance gain by employing extra neutral words is less significant.

Chapter 6

Conclusion and Future Work

In recent years, there has been a rapid growth of research interest in natural language processing that seeks to better understand sentiment or opinion expressed in text. One reason is that with the rise of various types of social media, communicating on the web has become increasingly popular, where millions of people broadcast their thoughts and opinions on a great variety of topics, such as feedback on products and services, opinions on political development and events, and information sharing on global disasters. Therefore, new computational tools are needed to help organize, summarize and understand this vast amount of information. Additionally, the discovery of opinions reflecting people's attitudes towards various topics enables many useful applications, which is another motivation of sentiment analysis.

Despite the recent successes, the field of sentiment analysis is still relatively new and there remains much to be explored. In this thesis, we have focused on the document-level sentiment classification and the sentence-level subjectivity detection, which are two main tasks of sentiment analysis. Most of the existing approaches to these two tasks rely on supervised or semi-supervised models trained from labelled data, where such labelled data may not be easy to obtain in real world applications. Another absence from most of the previous work is the consideration of dependencies between sentiment/subjectivity and topics. By modelling such dependencies, it may not only help find better feature representations for sentiment classification and subjectivity detection, but also can provide more informative sentiment-topic mining results to users.

This thesis presented three new probabilistic topic models, which address the above shortcomings of the current sentiment analysis approaches by modelling sentiment/subjectivity in conjunction with topics from text data. These new models are summarized as follows:

- The first model, JST, extends latent Dirichlet allocation (LDA) by constructing an additional sentiment layer, which detects sentiment and topic simultaneously from text. A mechanism is introduced to incorporate the prior information of sentiment lexicons into model learning by modifying the Dirichlet priors of the per-corpus word distributions. A reparameterized version of the JST model called Reverse-JST, obtained by reversing the sequence of sentiment and topic generation in the generative process, is also studied. Although JST and Reverse-JST are fundamentally the same without a hierarchical prior, extensive experiments show that when sentiment priors are added, JST performs consistently better than Reverse-JST. Besides, unlike supervised approaches to sentiment classification which often fail to produce satisfactory performance when applied to other domains, the weakly-supervised nature of JST makes it highly portable to other domains. This is verified by the experimental results on datasets from five different domains, where the JST model even outperforms the existing semi-supervised approaches [Li et al., 2009] in some of the datasets despite using no labelled documents. Moreover, the sentiment-bearing topics detected by JST are clearly interpretable.

- The second model, dJST, permits discovering and tracking the intimate interplay between sentiment and topic over time from data. The sentiment and topic dynamics are modelled by assuming that the current sentiment-topic word distributions are generated from the Dirichlet distributions parameterized by the word distributions of the documents from past epochs. To efficiently estimate the model parameters for a large corpus, we derived online inference procedures based on a stochastic EM algorithm, from which the dJST model can be updated with the previously estimated model and the newly arrived data. We compared dJST with two non-dynamic versions of JST in terms of predictive perplexity and sentiment classification accuracy, based on the Mozilla add-on review dataset crawled from 2007 to 2011. These two models are JST-one, which only uses the current data for training, and JST-all, which uses all the past data for model learning. Experimental results showed that dJST outperforms JST-one in both predictive perplexity and sentiment classification accuracy, which demonstrate the effectiveness of modelling dynamics. While JST-all achieves slightly better sentiment accuracy than dJST, the perplexity of dJST is much lower. Additionally, by avoiding the modelling of all the past documents, the computational time of dJST is in a small fraction of JST-all. Besides, the evolution of sentiment-bearing topics detected by dJST is indeed coherent and informative, which can be used as succinct summaries of document archives that collected over a large time span.

- The third model, subjLDA, tackles sentence-level subjectivity detection, which can automatically identify whether a given sentence expresses opinion or states facts. In contrast to most of the existing methods relying on either labelled corpora or linguistic pattern extraction for subjectivity classifier training, we view the problem as weakly-supervised generative model learning, where the only supervision information required is from a small set of domain independent subjectivity lexical clues. The subjLDA model has been evaluated on the Multi-Perspective Question Answering (MPQA) dataset. Experimental results show that, except slightly lower in objective recall, subjLDA outperforms LDA in all other evaluation metrics, and is comparable to the previously proposed bootstrapping approach [Wiebe and Riloff, 2005] which used a much larger dataset for training. We have also explored adding neutral words as prior information for model learning. It was found that while a significant performance gain can be achieved by incorporating subjectivity clues bearing positive or negative polarity, the prior lexical information from neutral words is less effective.

The new probabilistic topic models presented in this thesis address the needs for analysing large opinionated document archives. By modeling sentiment/subjectivity and topic simultaneously, the new models overcome some of the limitations of the current sentiment analysis approaches, and demonstrate the importance of the research direction pursued in the thesis with the useful results offered by the models.

6.1 Future Work

The current research work may be extended in several ways as described below.

6.1.1 Modelling Linguistic Knowledge

In most of the experiments of the JST, dJST and subjLDA models, documents are represented in the bag-of-words format, with word order being ignored. While this assumption is reasonable for uncovering the semantic structures of texts, word order is certainly important for sentiment analysis. For instance, it has been observed in our experiments that JST using a combination of uigram and bigram features achieved better sentiment accuracy than using either type of features alone. Therefore, it is worth modelling deeper linguistic knowledge such as syntactic structure of documents in order to improve the models' performance in sentiment classification and subjectivity detection.

6.1.2 Incorporating Other Types of Prior Information

The JST and dJST models only incorporate prior knowledge from sentiment lexicons for model learning. It is also possible to incorporate other types of prior information, such as some known topical knowledge of product reviews, for discovering more salient topics about product features and aspects. Another possibility is to develop a semi-supervised version of JST and dJST, with some supervised information being incorporated into the model parameter estimation procedure, such as use of the sentiment labels of reviews derived automatically from the ratings provided by users, to control the Dirichlet priors of the sentiment distributions.

6.1.3 Automatic Estimation of Topic Number

The models presented in this thesis were developed based on LDA, which assumes that the number of topics is known and fixed. In order to determine the optimal value for setting the number of topics, a typical procedure is to evaluate the models on a held-out dataset with respect to the predictive perplexity value. Such a process usually needs to be repeated when the models are applied to a dataset from a different domain. So another future direction would be to extend JST and dJST to a hierarchical Dirichlet process framework [Teh et al., 2006], which allows the number of topics to be inferred from data automatically.

6.1.4 Visualization and User Interfaces

One of the most common ways to display a topic is to list the topic words that have the highest probability for each topic, without any means for interaction or manipulation of the results. Therefore, it would be very helpful to develop methods that can visualize the sentiment-bearing topics detected by JST and dJST, and allow interacting with the results for better exploring the structures and sentiments of large archives of documents.

Appendix A

The Fixed-Point Iteration Algorithm for Updating $\boldsymbol{\alpha}$

In the JST model, $\boldsymbol{\theta}_{d,l}$ is the parameter notation for the topic mixing proportions for document d and sentiment label l ($T-$ vector). For D documents and the sentiment label l, $\boldsymbol{\Theta}_l = \{\{\theta_{d,l,z}\}_{z=1}^{T}\}_{d=1}^{D}$ ($D \times T$ matrix). According to the JST model generative process, $\boldsymbol{\theta}_{d,l}$ is drawn from a Dirichlet distribution parameterized by $\boldsymbol{\alpha}_l = \{\alpha_{l,z}\}_{z=1}^{T}$, which can be formally defined as

$$
\begin{aligned}
P(\boldsymbol{\theta}_{d,l}) &\sim \mathrm{Dir}(\boldsymbol{\alpha}_l) \\
P(\boldsymbol{\theta}_{d,l}) &= \frac{\Gamma(\sum_{z=1}^{T}\alpha_{l,z})}{\prod_{z=1}^{T}\Gamma(\alpha_{l,z})}\prod_{z=1}^{T}\theta_{d,l,z}^{\alpha_{l,z}-1}
\end{aligned} \tag{A.1}
$$

where $\theta_{d,l,z} > 0$ and $\sum_{z=1}^{T}\theta_{d,l,z} = 1$.

Our objective is to estimate $\boldsymbol{\alpha}_l$ given the observations $\boldsymbol{\Theta}_l = \{\boldsymbol{\theta}_{1,l}, \boldsymbol{\theta}_{2,l} ... \boldsymbol{\theta}_{D,l}\}$, which maximizes the likelihood

$$
P(\boldsymbol{\Theta}_l|\boldsymbol{\alpha}_l) = \prod_{d=1}^{D} P(\boldsymbol{\theta}_{d,l}|\boldsymbol{\alpha}_l). \tag{A.2}
$$

Taking the log likelihood of Equation A.2, it then becomes

$$
\log P(\boldsymbol{\Theta}_l|\boldsymbol{\alpha}_l) = D[\log\Gamma(\sum_{z=1}^{T}\alpha_{l,z}) - \sum_{z=1}^{T}\log\Gamma(\alpha_{l,z}) + \sum_{z=1}^{T}(\alpha_{l,z}-1)\log\bar{\boldsymbol{\theta}}_{l,z}] \tag{A.3}
$$

where $\log\bar{\boldsymbol{\theta}}_{l,z} = \frac{1}{D}\sum_{d=1}^{D}\log\theta_{d,l,z}$.

Differentiating the log likelihood with respect to $\alpha_{l,z}$ gives:

$$
\frac{\partial P(\boldsymbol{\Theta}_l|\boldsymbol{\alpha}_l)}{\partial\,\alpha_{l,z}} = D\Psi(\sum_{z=1}^{T}\alpha_{l,z}) - D\sum_{z=1}^{T}\Psi(\alpha_{l,k}) + D\log\bar{\boldsymbol{\theta}}_{l,z}, \tag{A.4}
$$

where $\Psi(\cdot)$ is the digamma function defined by

$$\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x}. \tag{A.5}$$

Setting the differentiation to 0 yields

$$\Psi(\sum_{z=1}^{T} \alpha_{l,z}) = \sum_{z=1}^{T} \Psi(\alpha_{l,k}) + \log \bar{\boldsymbol{\theta}}_{l,z} \,. \tag{A.6}$$

Unfortunately, Equation A.6 doesn't provide a closed-from for calculating $\alpha_{l,z}$, and we therefore resort to a fixed-point iteration method to estimate $\alpha_{l,z}$.

The fixed-point iteration algorithm is derived as follows. Given an intimal guess for $\boldsymbol{\alpha}_l$, a lower bound on the likelihood is constructed which is tight at $\boldsymbol{\alpha}_l$. The bound is defined as [Minka, 2003]:

$$\Gamma(x) \geq \Gamma(\hat{x}) e^{((x-\hat{x})\Psi(\hat{x}))} \tag{A.7}$$

$$\log \Gamma(x) \geq \log \Gamma(\hat{x}) + (x - \hat{x})\Psi(\hat{x}), \tag{A.8}$$

where x is the true value and $\hat{x}$ is the estimated value.

Applying Equation A.8 to the first term on the RHS of Equation A.3 yields

$$\begin{aligned}
\log P(\boldsymbol{\Theta}_l|\boldsymbol{\alpha}_l) \geq{}& D\,[\log \Gamma(\sum_{z=1}^{T} \alpha_{l,z}^{old}) + (\sum_{z=1}^{T} \alpha_{l,z} - \sum_{z=1}^{T} \alpha_{l,z}^{old}) \Psi(\sum_{z=1}^{T} \alpha_{l,z}^{old})\,] - \\
& D \sum_{z=1}^{T} \log \Gamma(\alpha_{l,z}) + D \sum_{z=1}^{T} (\alpha_{l,z} - 1) \log \bar{\boldsymbol{\theta}}_{l,z} \qquad \text{(A.9)} \\
\geq{}& D \sum_{z=1}^{T} \alpha_{l,z} \Psi(\sum_{z=1}^{T} \alpha_{l,k}^{old}) - D \sum_{z=1}^{T} \log \Gamma(\alpha_{l,z}) + D \sum_{z=1}^{T} (\alpha_{l,z} - 1) \log \bar{\boldsymbol{\theta}}_{l,z} + \\
& \underbrace{D\,\Gamma(\sum_{z=1}^{T} \alpha_{l,z}^{old}) - D \sum_{z=1}^{T} \alpha_{l,z}^{old} \Psi(\sum_{z=1}^{T} \alpha_{l,z}^{old})}_{\text{(const.)}} \,. \qquad \text{(A.10)}
\end{aligned}$$

Differentiating Equation A.10 with respect to $\alpha_{l,z}$ gives:

$$\frac{\partial P(\boldsymbol{\Theta}_l|\boldsymbol{\alpha}_l)}{\partial \alpha_{l,z}} \geq D\,\Psi(\sum_{z=1}^{T} \alpha_{l,z}^{old}) - D\,\Psi(\alpha_{l,z}) + D \log \bar{\boldsymbol{\theta}}_{l,z} \tag{A.11}$$

Setting the differentiation to 0 gives:

$$\Psi(\alpha_{l,z}) = \underbrace{\Psi(\sum_{z=1}^{T} \alpha_{l,z}^{old}) + \log \bar{\boldsymbol{\theta}}_{l,z}}_{y} \,. \tag{A.12}$$

Finally, applying the inverted function of $\Psi(\cdot)$ yields

$$\alpha_{l,z} = \Psi^{-1}(y). \tag{A.13}$$

Using Equations A.12 and A.13, $\alpha_{l,z}$ can be calculated in the closed-form, and is guaranteed to converge to a stationary point of the likelihood after sufficient iterations.

Appendix B

Estimating the Weight Vector $\boldsymbol{\mu}^t$ 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], where the joint distribution is

$$P(\mathbf{w}^t, \mathbf{l}^t, \mathbf{z}^t | \mathbf{E}^{t-1}, \boldsymbol{\mu}^t, \boldsymbol{\alpha}^t, \gamma^t) = P(\mathbf{l}^t | \gamma^t) P(\mathbf{z}^t | \mathbf{l}^t, \boldsymbol{\alpha}^t) P(\mathbf{w}^t | \mathbf{l}^t \mathbf{z}^t, \mathbf{E}^{t-1}, \boldsymbol{\mu}^t). \tag{B.1}$$

We only need to focused the third term on the RHS of the joint distribution B.1 as it is the only term that contains $\boldsymbol{\mu}^t$:

$$P(\mathbf{w}^t | \mathbf{l}^t, \mathbf{z}^t, \mathbf{E}^{t-1}, \boldsymbol{\mu}^t) = \prod_{l=1}^{L} \prod_{z=1}^{T} \frac{\Gamma(\sum_s \mu^t_{l,z,s})}{\prod_{w=1}^{V} \Gamma(\sum_s \mu^t_{l,z,s} \sigma^{t-1}_{l,z,s,w})} \frac{\prod_{w=1}^{V} \Gamma(N^t_{l,z,w} + \sum_s \mu^t_{l,z,s} \sigma^{t-1}_{l,z,s,w})}{\Gamma(N^t_{l,z} + \sum_s \mu^t_{l,z,s})}. \tag{B.2}$$

Taking the log likelihood gives:

$$\begin{aligned} \log P(\mathbf{w}^t | \mathbf{l}^t, \mathbf{z}^t, \mathbf{E}^{t-1}, \boldsymbol{\mu}^t) = {} & \sum_{l=1}^{L} \sum_{z=1}^{T} \underbrace{[\log \Gamma(\sum_{z=1}^{T} \mu^t_{l,z,s}) - \log \Gamma(N^t_{l,z} + \sum_{s} \mu^t_{l,z,s})]}_{T1} + \\ & \sum_{l=1}^{L} \sum_{z=1}^{T} \sum_{w=1}^{V} \underbrace{[\log \Gamma(N^t_{l,z,w} + \sum_{s=1}^{S} \mu^t_{l,z,s} \sigma^{t-1}_{l,z,s,w}) - \log \Gamma(\sum_{s=1}^{S} \mu^t_{l,z,s} \sigma^{t-1}_{l,z,s,w})]}_{T2}, \end{aligned} \tag{B.3}$$

Term T1 and T2 in Equation B.3 can be bounded using the bound [Wallach, 2008]

$$\begin{aligned} \log \Gamma(z) - \log \Gamma(z+n) \geq {} & \\ & \log \Gamma(\hat{z}) - \log \Gamma(\hat{z}+n) + [\Psi(\hat{z}+n) - \Psi(\hat{z})](\hat{z} - z) \end{aligned} \tag{B.4}$$

and the bound [Wallach, 2008]

$$\begin{aligned}&\log\Gamma(z+n) - \log\Gamma(z) \geq \\ &\qquad \log\Gamma(\hat{z}+n) - \log\Gamma(\hat{z}) + \hat{z}[\Psi(\hat{z}+n) - \Psi(\hat{z})](\log z - \log\hat{z}).\end{aligned} \tag{B.5}$$

Applying bounds B.4 and B.5 to Equation B.3 yields

$$\begin{aligned}&\log P(\mathbf{w}^t|\mathbf{l}^t, \mathbf{z}^t, \mathbf{E}^{t-1}, \boldsymbol{\mu}^t) \geq \\ &\quad \sum_{l=1}^{L}\sum_{z=1}^{T}\{\log\Gamma(\sum_{s'=1}^{S}\mu^t_{l,z,s'}) - \log\Gamma(N^t_{l,z} + \sum_{s'=1}^{S}\mu^t_{l,z,s'}) + \\ &\quad [\Psi(N^t_{l,z} + \sum_{s'=1}^{S}\mu^t_{l,z,s'}) - \Psi(\sum_{s'=1}^{S}\mu^t_{l,z,s'})]\cdot(\sum_{s'=1}^{S}\mu^S_{l,z,s'} - \sum_{s=1}^{S}\mu^t_{l,z,s})\} + \\ &\quad \sum_{l=1}^{L}\sum_{z=1}^{T}\sum_{w=1}^{V}\{\log\Gamma(N^t_{l,z,w} + \sum_{s'=1}^{S}\mu_{l,z,s'}\sigma^{t-1}_{l,z,s',w}) - \log\Gamma(\sum_{s'=1}^{S}\mu_{l,z,s'}\sigma^{t-1}_{l,z,s',w}) + \\ &\quad \sum_{s'=1}^{S}\mu_{l,z,s'}\sigma^{t-1}_{l,z,s',w}[\Psi(N^t_{l,z,w} + \sum_{s'=1}^{S}\mu^t_{l,z,s'}\sigma^{t-1}_{l,z,s',w}) - \Psi(\sum_{s'=1}^{S}\mu^t_{l,z,s'}\sigma^{t-1}_{l,z,s',w})]\cdot \\ &\quad [\underbrace{\log(\sum_{s=1}^{S}\mu^t_{l,z,s}\sigma^{t-1}_{l,z,s,w})}_{T3} - \log(\sum_{s'=1}^{S}\mu^t_{l,z,s'}\sigma^{t-1}_{l,z,s',w})]\},\end{aligned} \tag{B.6}$$

where term T3 in Equation B.6 can be further bounded using the following bound

$$\log(a+b) \geq \log a + \log b, \tag{B.7}$$

giving

$$\log(\sum_{s=1}^{S}\mu^t_{l,z,s}\sigma^{t-1}_{l,z,s,w}) \geq \sum_{s=1}^{S}(\log\mu^t_{l,z,s} + \log\sigma^{t-1}_{l,z,s,w}). \tag{B.8}$$

Differentiating Equation B.6 with respect to $\mu_{l,z,s}$ gives:

$$\begin{aligned}&\frac{\partial\log P(\mathbf{w}^t|\mathbf{l}^t, \mathbf{z}^t, \mathbf{E}^{t-1}, \boldsymbol{\mu}^t)}{\partial\,\mu^t_{l,z,s}} \\ &\quad \geq -\underbrace{[\Psi(N^t_{l,z} + \sum_{s'=1}^{S}\mu^t_{l,z,s'}) - \Psi(\sum_{s'=1}^{S}\mu_{l,z,s'})]}_{B^t_{l,z}} + \\ &\sum_{w=1}^{V}\sum_{s'=1}^{S}\mu^t_{l,z,s'}\sigma^{t-1}_{l,z,s',w}\underbrace{[\Psi(N^t_{l,z,w} + \sum_{s'=1}^{S}\mu^t_{l,z,s'}\sigma^{t-1}_{l,z,s',w}) - \Psi(\sum_{s'=1}^{S}\mu^t_{l,z,s'}\sigma^{t-1}_{l,z,s',w}))]}_{A^t_{l,z,w}}\cdot\frac{1}{\mu_{l,z,s}}\end{aligned} \tag{B.9}$$

Setting the differentiation to 0 gives:

$$(\mu_{l,z,s}^{t})^{new} = \frac{\mu_{l,z,s'} \sum_{w=1}^{V} \sigma_{l,z,s',w}^{t-1} \cdot A_{l,z,w}^{t}}{B_{l,z}^{t}} \tag{B.10}$$

References

A. Abbasi, H. Chen, and A. Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Transactions on Information Systems (TOIS)*, 26(3):1–34, 2008. 2.1.3

A. Andreevskaia and S. Bergler. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of the Association for Computational Linguistics and the Human Language Technology Conference (ACL-HLT)*, pages 290–298, 2008. 2.1.3

D. Andrzejewski, X. Zhu, M. Craven, and B. Recht. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Barcelona, Spain*, 2011. 2.2.2, 2.2.5

S. Argamon, K. Bloom, A. Esuli, and F. Sebastiani. Automatically determining attitude type and force for sentiment analysis. *Human Language Technology. Challenges of the Information Society*, pages 218–231, 2009. 2.1.3

A. Aue and M. Gamon. Customizing sentiment classifiers to new domains: a case study. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, 2005. 1.3.1, 2.1.2.1

C. Banea, R. Mihalcea, J. Wiebe, and S. Hassan. Multilingual subjectivity analysis using machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 127–135, 2008. 2.1.2.2

C.M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006. 2.1.5, 2.2.1, 2.2.1.1

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003. ISSN 1533-7928. 1, 1.2, 2.2, 2.2, 2.2.1, 2.2.5, 3.1, 5.1, 5.4.1

D.M. Blei. Introduction to probabilistic topic models. 2011. 1.2

D.M. Blei and J.D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, pages 113–120, 2006. 1.2, 2.2.4, 2.2.5

D.M. Blei and J.D. McAuliffe. Supervised topic models. *Arxiv preprint arXiv:1003.0783*, 2010. 2.2.2, 2.2.5

J Blitzer, M Dredze, and F Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 440–447, 2007. 1.3.1, 2.1.2.1, 3.4.1

J. Bollen, A. Pepe, and H. Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. 2009. 2.1.4.1

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. 2.1.4.1

J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2011. 1.1, 1.2

J. Boyd-Graber and D. Blei. Putop: Turning predominant senses into a topic model for word sense disambiguation. In *Proceedings of the International Workshop on Semantic Evaluations (SemEval)*, pages 277–281, 2007. 2.2.2

J. Boyd-Graber and D.M. Blei. Syntactic topic models. *Arxiv preprint arXiv:1002.4665*, 2010. 2.2.2

S. Brody and N. Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 804–812, 2010. 2.2.3

S. Dasgupta and V. Ng. Mine the easy, classify the hard: a semi-supervised approach to automatic sentiment classification. In *Proceedings of the Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 701–709, 2009a. 2.1.2.1

S. Dasgupta and V. Ng. Topic-wise, sentiment-wise, or otherwise? identifying the hidden dimension for unsupervised text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 580–589, 2009b. 2.1.3, 3.5.2, 3.5.3

K. Duh, A. Fujino, and M. Nagata. Is machine translation ripe for cross-lingual sentiment classification? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 429–433, Portland, Oregon, USA, 2011. 2.1.2.2

K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 345–354, 2006. 2.2.3

E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5220, 2004. 2.2.2, 2.2.5

A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422, 2006. 5.1

L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 524–531, 2005. 2.2.5

T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228, 2004. 1.2, 2.2.1

T.L. Griffiths, M. Steyvers, D.M. Blei, and J.B. Tenenbaum. Integrating topics and syntax. *Advances in Neural Information Processing Systems*, 17:537–544, 2005. 2.2.2

Y. He, C. Lin, and H. Alani. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 123–131, 2011. 2.1.2.1, 2.2.3

G. Heinrich. Parameter estimation for text analysis. *Web: http://www. arbylon. net/publications/textest*, 2005. 2.2.1, 4.3.2

M.D. Hoffman, D.M. Blei, and F. Bach. Online learning for latent Dirichlet allocation. *Advances in Neural Information Processing Systems (NIPS)*, 23:856–864, 2010. 1.2, 2.2.1

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 50–57, 1999. 1.2, 2.2, 2.2, 2.2.3

T. Iwata, T. Yamada, Y. Sakurai, and N. Ueda. Online multiscale dynamic topic models. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 663–672, 2010. 2.2.4, 2.2.5, 4.2, 4.2.2.1

Y. Jo and A.H. Oh. Aspect and sentiment unification model for online review analysis. In *Proceedings of the International Conference on Web Search and Data Mining*, pages 815–824, 2011. 2.2.3

M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999. 2.2.1

N. Kaji and M. Kitsuregawa. Automatic construction of polarity-tagged corpus from html documents. In *Proceedings of the COLING-ACL*, pages 452–459, 2006. 2.1.3

H. Kanayama and T. Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 355–363, 2006. 2.1.3

A. Kennedy and D. Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, 22(2):110–125, 2006. 1.3.1, 1.3.2

S. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, page 1367, 2004. 1.1, 1.2, 2.1.3

S. Lacoste-Julien, F. Sha, and M.I. Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. *Advances in Neural Information Processing Systems (NIPS)*, 21, 2008. 3.2.1

S. Li and C. Zong. Multi-domain sentiment classification. In *Proceedings of the Association for Computational Linguistics and the Human Language Technology Conference (ACL-HLT)*, pages 257–260, 2008. 2.1.2.1

S. Li, C.R. Huang, G. Zhou, and S.Y.M. Lee. Employing personal/impersonal views in supervised and semi-supervised sentiment classification. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 414–423, 2010. 2.1.2.1

T. Li, Y. Zhang, and V. Sindhwani. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *Proceedings of the Joint conference of the Annual Meeting of the Association for Computational Linguistics and the*

International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP), pages 244–252, 2009. 2.1.2.1, 3.5, 3.5.2, 3.5.3, 6

C Lin and Y He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the ACM conference on Information and knowledge management (CIKM)*, 2009. 2.2.3, 3.1, 3.2

C. Lin, Y. He, and R. Everson. A comparative study of Bayesian models for unsupervised sentiment detection. In *Proceedings of Conference on Computational Natural Language Learning (CoNLL)*, 2010. 2.2.3

C. Lin, Y. He, and R. Everson. Sentence subjectivity detection with weakly-supervised learning. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, 2011a. 2.1.4.2

C. Lin, Y. He, R. Everson, and S. Rüger. Weakly-Supervised Joint Sentiment-Topic Detection from Text. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2011b. ISSN 1041-4347. 2.1.2.1, 2.2.3, 3.2, 4.1

B. Lu, C. Tan, C. Cardie, and Benjamin K. Tsou. Joint bilingual sentiment classification with unlabeled parallel corpora. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 320–330, Portland, Oregon, USA, 2011. 2.1.2.2

T. Lux. Sentiment dynamics and stock returns: The case of the german stock market. Kiel working papers, Kiel Institute for the World Economy, 2008. 2.1.4.1

Y. Mao and G. Lebanon. Isotonic conditional random fields and local sentiment flow. *Advances in neural information processing systems*, 19:961, 2007. 2.1.4.1

Y. Mao and G. Lebanon. Generalized isotonic conditional random fields. *Machine learning*, 77(2):225–248, 2009. 2.1.4.1

B. Marlin. Modeling user rating profiles for collaborative filtering. volume 16, pages 627–634, 2004. 2.2.5

S. Matsumoto, H. Takamura, and M. Okumura. Sentiment classification using word subsequences and dependency sub-trees. In *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-05)*, pages 301–310, 2005. 2.1.1

A. McCallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 786–791, 2005. 2.2.2, 2.2.5

A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal of Artificial Intelligence Research*, 30(1):249–272, 2007. 1.2

R. McDonald, K. Hannan, T. Neylon, and J. Wells, M .and Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 432–439, 2007. 2.1.1

D.M. McNair, M. Lorr, and L.F. Droppleman. Profile of mood states (poms). *San Diego*, 1971. 2.1.4.1

Q Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web (WWW)*, pages 171–180, 2007. 2.1.4.1, 2.2.3, 3.1

P. Melville, W. Gryc, and R.D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1275–1284, 2009. 2.1.3

R. Mihalcea, C. Banea, and J. Wiebe. Learning multilingual subjective language via cross-lingual projections. In *In Proceedings of the Association for Computational Linguistics (ACL)*, page 976, 2007. 1.1, 2.1.2.2, 2.1.4.2

D. Mimno and A. McCallum. Organizing the OCA: learning faceted subjects from a library of digital books. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 376–385, 2007. 1.2

D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *Uncertainty in Artificial Intelligence*, pages 411–418. Citeseer, 2008. 2.2.2, 2.2.5

T. Minka. Estimating a Dirichlet distribution. Technical report, MIT, 2003. 3.4.3, 4.1, 4.2.2, 4.2.2.2, 4.2.3, 5.3.3, A, B

G. Murray and G. Carenini. Predicting subjectivity in multimodal conversations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1348–1357, 2009. 1.3.4, 2.1.4.2, 5.1

R.M. Nallapati, S. Ditmore, J.D. Lafferty, and K. Ung. Multiscale topic tomography. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 520–529, 2007. 2.2.4, 4.2

B. OConnor, R. Balasubramanyan, B.R. Routledge, and N.A. 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*, pages 122–129, 2010. 1.1, 1.2

S.J. Pan, X. Ni, J.T. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 751–760, 2010. 2.1.2.1

B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, page 271, 2004. 1.3.1, 1.3.2, 2.1.1, 2.1.4.2, 3.4.1, 3.5.2

B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2 (1-2):1–135, 2008. 1, 1.1, 1.3.4, 2.1.5, 3.2.1

B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002. 1.1, 1.2, 1.3.1, 1.3.2, 2.1.1, 2.2.3, 3.4.1

M.F. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 40(3):211–218, 2006. 3.4.1, 4.3.1

I. Pruteanu-Malinici, L. Ren, J. Paisley, E. Wang, and L. Carin. Hierarchical Bayesian modeling of topics in time-stamped documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 996–1011, 2009. 2.2.4

L. Qiu, W. Zhang, C. Hu, and K. Zhao. SELC: a self-supervised model for sentiment classification. In *Proceeding of the 18th ACM conference on Information and knowledge management (CIKM)*, pages 929–936, 2009. 2.1.3

S. Raaijmakers, K. Truong, and T. Wilson. Multimodal subjectivity analysis of multiparty conversation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 466–474, 2008. 1.3.4, 2.1.4.2, 5.1

D. Ramage, D. Hall, R. Nallapati, and C.D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 248–256, 2009. 3.2.1

J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48, 2005. 2.1.5

J. Read and J. Carroll. Weakly supervised techniques for domain-independent sentiment classification. In *Proceeding of the 1st International CIKM Workshop on Topic-sentiment Analysis for Mass Opinion*, pages 45–52, 2009. 2.1.3

L. Ren, D.B. Dunson, and L. Carin. The dynamic hierarchical Dirichlet process. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 824–831, 2008. 2.2.4

E. Riloff and J. Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the conference on Empirical methods in natural language processing (EMNLP)*, pages 105–112, 2003. 1.3.4, 2.1.4.2, 5.3.2

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 487–494, 2004. 2.2.2, 2.2.5

M. Steyvers and T. Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427, 2007. 2.2.1, 2.2.1, 3.4.3, 4.2.3, 5.3.3

S. Tan, G. Wu, H. Tang, and X. Cheng. A novel scheme for domain-transfer problem in the context of sentiment analysis. In *Proceedings of the ACM conference on Conference on Information and Knowledge Management (CIKM)*, pages 979–982, 2007. 2.1.2.1

S. Tan, Y. Wang, and X. Cheng. Combining learn-based and lexicon-based techniques for sentiment detection without using labeled examples. In *Proceedings of the International ACM Conference on Research and Development in Information Eetrieval (SIGIR)*, pages 743–744, 2008. 2.1.3

Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006. 2.2.4, 6.1.3

I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceeding of the International Conference on World Wide Web (WWW)*, pages 111–120, 2008a. 2.2.3

Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the Aunal Meeting on Association for Computational Linguistics and the Human Language Technology Conference (ACL-HLT)*, pages 308–316, 2008b. 2.2.3, 3.1

P. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Annual Meeting on Association for Computational Linguistics (ACL)*, pages 417–424, 2001. doi: http://dx.doi.org/10.3115/1073083.1073153. 1.3.1

P. Turney and M Littman. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. *ArXiv Computer Science e-prints*, cs.LG/0212012, 2002. 1.1, 1.2, 2.1.3

H. Wallach, D. Mimno, and A. McCallum. Rethinking lda: Why priors matter. volume 22, pages 1973–1981, 2009. 3.4.3, 4.2.3

H.M. Wallach. *Structured topic models for language*. PhD thesis, University of Cambridge, 2008. B, B

X. Wan. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 235–243, 2009. 2.1.2.2

C. Wang, D. Blei, and D. Heckerman. Continuous time dynamic topic models. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 2009. 2.2.4, 2.2.5

X. Wang and A. McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 424–433, 2006. 2.2.3, 2.2.4, 2.2.5

X. Wang, N. Mohanty, and A. McCallum. Group and topic discovery from relations and text. In *Proceedings of the 3rd International Workshop on Link Discovery*, pages 28–35, 2005. 2.2.2

C. Whitelaw, N. Garg, and S. Argamon. Using appraisal groups for sentiment analysis. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, pages 625–631, 2005. 1.3.1, 1.3.2, 2.1.1, 2.1.3

J. Wiebe and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, volume 3406, pages 486–497. Springer, 2005. 1.1, 1.3.4, 1.4, 2.1.4.2, 5.1, 5.3.2, 5.2, 5.4.1, 6

T. Wilson and S. Raaijmakers. Comparing word, character, and phoneme n-grams for subjective utterance recognition. In *Proceedings of INTERSPEECH*, pages 1614–1617, 2008. 1.3.4, 2.1.4.2, 5.1

A. Yessenalina, Y. Yue, and C. Cardie. Multi-level structured models for document-level sentiment classification. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1046–1056, 2010. 2.1.1

H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 129–136, 2003. 1.1, 1.2, 2.1.4.2, 5.1

T. Zagibalov and J. Carroll. Automatic seed word selection for unsupervised sentiment classification of chinese text. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 1073–1080, 2008a. 2.1.3

T. Zagibalov and J. Carroll. Unsupervised classification of sentiment and objectivity in chinese text. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, number 1, pages 304–311, 2008b. 2.1.3

O. Zaidan, J. Eisner, and C. Piatko. Using annotator rationales to improve machine learning for text categorization. In *Proceedings of NAACL-HLT*, pages 260–267, 2007. 2.1.1

J. Zhang, Y. Song, C. Zhang, and S. Liu. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1079–1088, 2010. 2.2.4

W.X. Zhao, J. Jiang, H. Yan, and X. Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 56–65, 2010. 2.2.3, 2.2.5