Joint Sentiment/Topic Model for Sentiment Analysis

Chenghua Lin
School of Engineering, Computing and Mathematics
University of Exeter
North Park Road, Exeter EX4 4QF, UK
cl322@exeter.ac.uk

Yulan He
Knowledge Media Institute
The Open University
Milton Keynes MK7 6AA, UK
y.l.he.01@cantab.net

ABSTRACT

Sentiment analysis or opinion mining aims to use automated tools to detect subjective information such as opinions, attitudes, and feelings expressed in text. This paper proposes a novel probabilistic modeling framework based on Latent Dirichlet Allocation (LDA), called joint sentiment/topic model (JST), which detects sentiment and topic simultaneously from text. Unlike other machine learning approaches to sentiment classification which often require labeled corpora for classifier training, the proposed JST model is fully unsupervised. The model has been evaluated on the movie review dataset to classify the review sentiment polarity and minimum prior information have also been explored to further improve the sentiment classification accuracy. Preliminary experiments have shown promising results achieved by JST.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms

Algorithms, Experimentation

Keywords

Sentiment analysis, Opinion mining, Latent Dirichlet Allocation, Joint sentiment/topic model

1. INTRODUCTION

As propelled by the rapid growth of text data, text mining has been applied to discover hidden knowledge from text in many applications and domains. In business sectors, great efforts have been made to find out customers' sentiments and opinions, often expressed in free text, towards companies' products and services. However, discovering sentiments and opinions through manual analysis of a large volume of textual data is extremely difficult. Hence, in recent years, there have been much interests in the natural language processing community to develop novel text mining techniques with the capability of accurately extracting customers' opinions from large volumes of unstructured text data.

Among various opinion mining tasks, one of them is sentiment classification, i.e. whether the semantic orientation of a text is positive, negative or neutral. When applying machine learning to sentiment classification, most existing approaches rely on supervised learning models trained from labeled corpora where each document has been labeled as positive or negative prior to training. Such labeled corpora are not always easily obtained in practical applications. Also, sentiment classification models trained on one domain might not work at all when moving to another domain. Furthermore, in a more fine-grained sentiment classification problem (e.g. finding users' opinions for a particular product feature), topic/feature detection and sentiment classification are often performed in a two-stage pipeline process, by first detecting a topic/feature and later assigning a sentiment label to that particular topic.

Intuitively, sentiment polarities are dependent on topics or domains. Therefore, detecting both topic and sentiment simultaneously should serve a critical function in helping users in terms of opinion mining and summarization. For instance, though the adjective 'unpredictable' in a phrase such as 'unpredictable steering' may have negative orientation in an automobile review, it could also have positive orientation in a phrase like 'unpredictable plot' in a movie review [5].

Although much work has been done in detecting topics [2, 6, 20], these lines of work mainly focused on discovering and analyzing topics of documents alone, without any analysis of sentiment in the text, which limit the usefulness of the mining results. Other work [16, 22, 11, 15, 4, 3, 25] addressed the problem of sentiment detection in various levels (i.e. from word/phrase level, to sentence and document level). However, none of them can model mixture of topics alongside with sentiment classification, which again makes the results less informative to users. Some of the recent work [14, 19] has been aware of this limitation and tried to capture sentiments and mixture of topics simultaneously. However, Mei *et al.* [14] does not model sentiment directly and requires post-processing to calculate the positive/negative coverage in a document in order to identify its polarity. Titov and McDonald [19] requires some kind of supervised settings that the customer reviews should contain ratings for the aspects/features discussed in the text and thus it lacks of the flexibility to adapt to other domains.

CIKM'09, November 2–6, 2009, Hong Kong, China.

In this paper, we focus on document level sentiment classification based on the proposed unsupervised joint sentiment/topic (JST) model. This model extends the state-of-the-art topic model, Latent Dirichlet Allocation (LDA), by adding a sentiment layer. Our model distinguishes from other models in that: (1) JST is fully unsupervised; (2) JST can detect sentiment and topic simultaneously. To the best of our knowledge, no other existing approaches present the same merits as our model. We have also explored various approaches for obtaining prior information in order to improve the sentiment detection accuracy. Although the proposed JST model can be easily extended to detect polarity of text at various granularity levels, in this paper we mainly focus on reporting our preliminary results on the document-level sentiment classification and briefly present the sentiment analysis results on some extracted topics as an example illustration.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 presents the Joint Sentiment/Topic (JST) model. We show the experimental setup in Section 4 and discuss the results based on the movie review dataset[1] in Section 5. Finally, Section 6 concludes the paper and outlines the future work.

2. RELATED WORK

Great bulk of work has been focused on the problem of sentiment classification at various levels using machine learning techniques. Turney and Littman [22] applied an unsupervised learning algorithm to classify the semantic orientation in the word/phrase level, based on mutual information between document phrases and a small set of positive/negative paradigm words like "good" and "bad". Choi *et al.* [4] dealt with opinion analysis by combining conditional random fields (CRFs) and a variation of Autoslog. In the sentence level, a semi-supervised machine learning algorithm was proposed by Pang and Lee [15], which employs a subjectivity detector and minimum cuts in graphs. Another system by Kim and Hovy [11] judges the sentiment of a given sentence by combining the individual word-level sentiment. Eguchi and Lavrenko [5] proposed a generative model that jointly models sentiment words, topic words and sentiment polarity in a sentence as a triple. In more recent work [25], the authors tackled this problem utilizing CRFs and considered both contextual dependency and label redundancy in sentence sentiment classification. Another line of work is in the document level, where one tries to evaluate the overall sentiment of a document. The representative work at the early stage can be found in [21, 16], where the former used unsupervised learning and mutual information, which is similar to the approach proposed in [22]; while the latter classified the polarity of movie reviews with the traditional supervised text categorization methods. Following this way, lots of other approaches have been proposed. For example, McDonald *et al.* [13] investigated a global structured model that learns to predict sentiment of different levels of granularity in text. Blitzer *et al.* [3] focused on domain adaption for sentiment classifiers with respect to different types of products' online reviews.

However, as can be easily pointed out, all the aforementioned work shares some similar limitations: (1) they only focus on sentiment classification without considering the mixture of topics in the text, which is less informative to users and may limit the usefulness of the results; (2) most of the approaches [16, 15, 4, 3, 13, 25] are favored in supervised learning, which require a labeled corpus for training and potentially restrain their applicability to other domains of interest.

Motivated by these observations, we construct an unsupervised hierarchical Bayesian model which can classify document level sentiment and extract mixture of topics simultaneously. To the best of our knowledge, not much work has been done regarding this particular problem. However, there are indeed several lines of work which are quite close to our vision [14, 20, 19].

One of the most closely related work is the Topic-Sentiment Model (TSM) [14], which jointly models the mixture of topics and sentiment predictions for the entire document. However, there are several intrinsical differences between JST and TSM. First of all, TSM is essentially based on the Probabilistic Latent Semantic Indexing (pLSI) [8] model with an extra background component and two additional sentiment subtopics, and thus suffers from the problems of inferencing on new document and overfitting the data, both of which are known as the deficits of pLSI. JST overcomes these shortcomings as it is based on LDA with a better statistical foundation. Regarding topic extraction, TSM samples a word either from the background component model or topical themes where the latter are further categorized into three sub-categories, i.e. neutral, positive and negative sentiment models. In contrast, in JST one draws a word from the distribution over words jointly defined by topic and sentiment label that chosen in the first place. Thirdly, 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 document.

Other models by Titov and McDonald [20, 19] are also closely related to ours, since they are all based on the state-of-the-art topic model LDA. First proposed in [20], the Multi-Grain Latent Dirichlet Allocation model (MG-LDA) is argued to be more appropriate to build topics that are representative of ratable aspects of objects from online user reviews, by allowing terms being 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) [19] 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 the MG-LDA. Our model differs from MAS in several aspects: MAS works on a supervised setting as it requires that every aspect is rated at least in some documents, which is practically infeasible in real life applications, while our JST model is fully unsupervised with only minimum prior information being incorporated, which in turn has more flexibilities; MAS focuses on extracting text for sentiment summaries of each aspect ratings while we predict the sentiment orientation in the document level.

3. JOINT SENTIMENT/TOPIC (JST) MODEL

The Latent Dirichlet Allocation (LDA) model, as shown

[1]Polarity dataset v2.0 URL: http://www.cs.cornell.edu/people/pabo/movie-review-data/

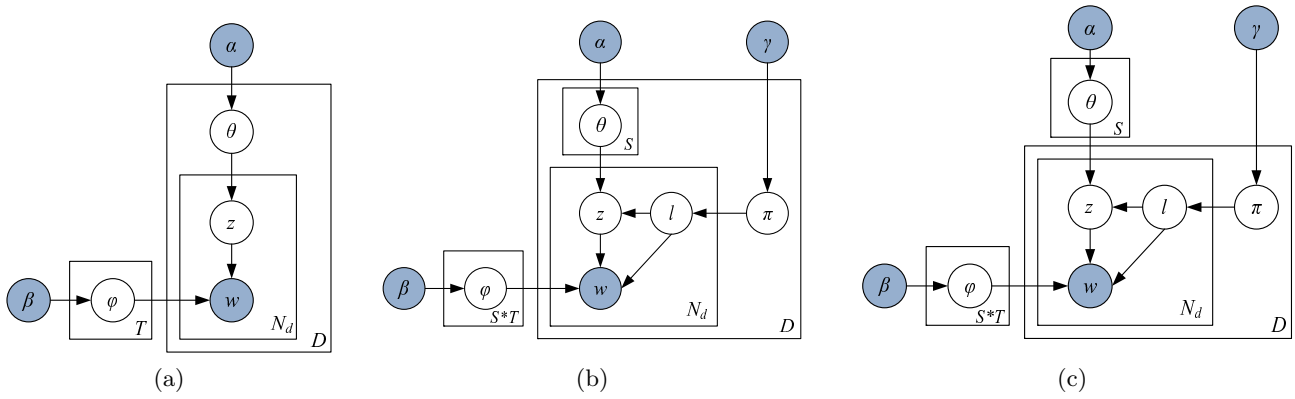

Figure 1: (a) LDA model; (b) JST model; (c) Tying-JST model.

in Figure 1(a), is one of the most popular topic models based upon the assumption that documents are mixture of topics, where a topic is a probability distribution over words [2, 18]. The LDA model is effectively a generative model from which a new document can be generated in a predefined probabilistic procedure. Compared to another commonly used generative model Probabilistic Latent Semantic Indexing (pLSI) [8], LDA has a better statistical foundation by defining the topic-document distribution θ, which allows inferencing on new document based on previously estimated model and avoids the problem of overfitting, where both are known as the deficits of pLSI. Generally, the procedure of generating each word in a document under LDA can be broken down into two stages. One firstly chooses a distribution over a mixture of K topics. Following that, one picks up a topic randomly from the topic distribution, and draws a word from that topic according to the topic's word probability distribution.

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 by adding an additional sentiment layer between the document and the topic layer. 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 1(b).

Assume that we have a corpus with a collection of D documents denoted by $C = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_D\}$; each document in the corpus is a sequence of N_d words denoted by $\mathbf{d} = (w_1, w_2, ..., w_{N_d})$, 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 procedure of generating a word w_i in document d boils down to three stages. Firstly, one chooses a sentiment label l from the document specific sentiment distribution π_d. Following that, one chooses a topic randomly from the topic distribution $\theta_{l,d}$, where $\theta_{l,d}$ is chosen conditioned on the sentiment label l. It is worth noting at this point that the topic-document distribution of JST is different from the one of LDA. In LDA, there is only one topic-document distribution θ for each individual document. In contrast, each document in JST is associated with S (number of sentiment labels) topic-document distributions, each of which corresponds to a sentiment label l with the same number of topics. This feature essentially provides means for the JST model to measure the sentiment of topics. Finally, one draws a word from distribution over words defined by the topic and sentiment label, which is again different from LDA that a word is sampled from the word distribution only defined by topic.

The formal definition of the generative process which corresponds to the hierarchical Bayesian model shown in Figure 1(b) is as follows:

- For each document d, choose a distribution $\pi_d \sim Dir(\gamma)$.
- For each sentiment label l under document d, choose a distribution $\theta_{d,l} \sim Dir(\alpha)$.
- For each word w_i in document d
  - choose a sentiment label $l_i \sim \pi_d$,
  - choose a topic $z_i \sim \theta_{d,l_i}$,
  - choose a word w_i from the distribution over words defined by the topic z_i and sentiment label l_i, $\varphi^{l_i}_{z_i}$.

The hyperparameters α and β in JST can be treated as the prior observation counts for the number of times topic j associated with sentiment label l sampled from a document and the number of times words sampled from topic j associated with sentiment label l respectively, 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 sampled from document before any words from the corpus is observed. In JST, there are three sets of latent variables that we need to infer, including: the joint sentiment/topic-document distribution θ, the joint sentiment/topic-word distribution φ, and the sentiment-document distribution π. We will see later in the paper that the sentiment-document distribution π plays an important role in determining the document polarity.

In order to obtain the distributions of θ, φ and π, we firstly estimate the posterior distribution over z, i.e the assignment of word tokens to topics and sentiment labels. The sampling distribution for a word given the remaining topics and sentiment labels is $P(z_t = j, l_t = k|\mathbf{w}, \mathbf{z}_{-\mathbf{t}}, \mathbf{l}_{-\mathbf{t}}, \alpha, \beta, \gamma)$ where $\mathbf{z}_{-\mathbf{t}}$ and $\mathbf{l}_{-\mathbf{t}}$ are vector of assignments of topics and

labels for all the words in the collection except for the word at position t in document d.

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

$$P(\mathbf{w}, \mathbf{z}, \mathbf{l}) = P(\mathbf{w}|\mathbf{z}, \mathbf{l})P(\mathbf{z}, \mathbf{l}) = P(\mathbf{w}|\mathbf{z}, \mathbf{l})P(\mathbf{z}|\mathbf{l}, d)P(\mathbf{l}|d) \quad (1)$$

For the first term, by integrating out φ, we obtain:

$$P(\mathbf{w}|\mathbf{z}, \mathbf{l}) = \left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V}\right)^{T*S} \prod_j \prod_k \frac{\prod_i \Gamma(N_{i,j,k} + \beta)}{\Gamma(N_{j,k} + V\beta)} \quad (2)$$

where V is the size of the vocabulary, T is the total number of topics, S is the total number of sentiment labels, $N_{i,j,k}$ is the number of times word i appeared in topic j and with sentiment label k. $N_{j,k}$ is the number of times words assigned to topic j and sentiment label k, and Γ is the gamma function.

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

$$P(\mathbf{z}|\mathbf{l}, d) = \left(\frac{\Gamma(T\alpha)}{\Gamma(\alpha)^T}\right)^{S*D} \prod_k \prod_d \frac{\prod_j \Gamma(N_{j,k,d} + \alpha)}{\Gamma(N_{k,d} + T\alpha)} \quad (3)$$

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

For the third term, by integrating out π, we obtain:

$$P(\mathbf{l}|d) = \left(\frac{\Gamma(S\gamma)}{\Gamma(\gamma)^S}\right)^D \prod_d \frac{\prod_k \Gamma(N_{k,d} + \gamma)}{\Gamma(N_d + S\gamma)} \quad (4)$$

where D is the total number of documents in the collection, $N_{k,d}$ is the number of times sentiment label k has been assigned to some word tokens in document d. N_d is the total number of words in the document collection.

Gibbs sampling will sequentially sample each variable of interest, z_t and l_t here, from the distribution over that variable given the current values of all other variables and the data. Letting the subscript $-t$ denote a quantity that excludes data from t^{th} position, the conditional posterior for z_t and l_t is:

$$P(z_t = j, l_t = k|\mathbf{w}, \mathbf{z}_{-t}, \mathbf{l}_{-t}, \alpha, \beta, \gamma) \propto \frac{\{N_{w_t,j,k}\}_{-t} + \beta}{\{N_{j,k}\}_{-t} + V\beta} \cdot \frac{\{N_{j,k,d}\}_{-t} + \alpha}{\{N_{k,d}\}_{-t} + T\alpha} \cdot \frac{\{N_{k,d}\}_{-t} + \gamma}{\{N_d\}_{-t} + S\gamma} \quad (5)$$

Equation 5 is the conditional probability derived by marginalizing out the random variables φ, θ, and π. A sample obtained from the Markov chain can be used to approximate the distribution of words in topics and sentiment labels:

$$\varphi_{i,j,k} = \frac{N_{i,j,k} + \beta}{N_{j,k} + V\beta} \quad (6)$$

The approximated predictive distribution over topics for sentiment label is:

$$\theta_{j,k,d} = \frac{N_{j,k,d} + \alpha}{N_{k,d} + T\alpha} \quad (7)$$

Finally, the approximated predictive distribution over sentiment label for document is:

$$\pi_{k,d} = \frac{N_{k,d} + \gamma}{N_d + S\gamma} \quad (8)$$

The pseudo code for the Gibbs sampling procedure of JST is shown in Figure 2.

1. Initialize $V \times T \times S$ matrix $\mathbf{\Phi}$, $T \times S \times D$ matrix $\mathbf{\Theta}$, $S \times D$ matrix $\mathbf{\Pi}$.
2. for $m = 1$ to M Gibbs sampling iterations do
3. Read a word i from a document
4. Calculate the probability of assigning word i to topic and sentiment label based on Equation 5.
5. Sample a topic j based on the estimated probability obtained in Step 3.
6. Sample a sentiment label k.
7. Update the matrix $\mathbf{\Phi}$, $\mathbf{\Theta}$, and $\mathbf{\Pi}$ with new sampling results.
8. Go to step 3 until all words have been processed.
9. end for

Figure 2: Gibbs sampling procedure.

3.1 Tying-JST Model

A variation of JST model is presented in Figure 1(c), namely tying-JST model. The major difference between tying-JST and JST model falls into that, in order to sample a word in a document during the generative process, one has to choose a topic-document distribution θ_d for every document under the JST model, whereas in tying-JST there is only one topic-document distribution θ which accounts for all the documents in the corpus. Therefore, during the Gibbs sampling procedure, rather than having a $\mathbf{\Theta}$ matrix with dimension $T \times S \times D$ as for JST, the $\mathbf{\Theta}$ matrix of tying-JST has only $T \times S$ dimension. As a result, the approximated predictive distribution over topics for sentiment label is different from Equation 7 and should be reformulated as:

$$\theta_{j,k} = \frac{N_{j,k} + \alpha}{N_k + T\alpha} \quad (9)$$

where T is the total number of topics, $N_{j,k}$ is the total number of times topic j is associated with sentiment label k, and N_k is total number of times that a word is associated with sentiment label k.

Experimental results will be presented in Section 5 to compare the performance of the JST and the tying-JST model.

4. EXPERIMENTAL SETUP

In this section, we present the experimental setup of document polarity classification and topic extraction based on the movie review dataset. This dataset consists of two categories of free format movie review texts, with their overall sentiment polarity labeled either positive or negative. However, one should note that we do not use any of the polarity label information of the dataset in our experiments but only for evaluating the performance of the JST model, as our model is fully unsupervised.

4.1 Preprocessing

Preprocessing was performed on the movie review data before the subsequent experiments. Firstly, punctuation, numbers and other non-alphabet characters were removed. Secondly, for the purpose of reducing the vocabulary size and addressing the issue of data sparseness, stemming was

performed using the Porter's stemmer algorithm [17]. Stop words were also removed based on a stop word list[2]. After preprocessing, the corpus contains 2000 documents and 627,317 words with 25,166 distinct terms.

4.2 Defining Model Priors

As has been pointed out by Pang *et al.* [16], the sentiment classification problem is somehow more challenging than the traditional topic-based classification, since sentiment can be expressed in a more subtle manner while topics can be identified more easily according to the co-occurrence of keywords. One of the directions for improving the sentiment detection accuracy is to incorporate prior information or subjectivity lexicon (i.e., words bearing positive or negative polarity), which can be obtained in many different ways. Some approach annotates polarity to words based on manually constructed Appraisal Groups [24]. Other approach generates subjectivity lexicons in a semi-automatic manner [1]. More recently, Kaji and Kitsuregawa [9] proposed a method which can build polarity-tagged corpus from HTML documents fully automatically. While subjectivity lexicon generation is beyond the scope of this paper, here in our experiments, we investigated incorporating prior information obtained in four different ways into the JST and the tying-JST model, and explored how the prior information can improve the sentiment classification accuracy.

Paradigm word list The paradigm word list consists of a set of positive and negative words, e.g. *excellent* and *rubbish*. These lexicon words can be simply treated as the paradigms for defining the positive and negative semantic orientation, rather than for the purpose of training the algorithm [22].

The majority of the words were derived from the word lists used by Pang *et al.* [16] for their baseline result tests, with punctuation like '?' and '!' removed. However, we did notice the difference that the movie review data used by Pang *et al.* [16] is an older version with only 700 positive and 700 negative movie reviews, compared to the newer version we used that contains 1000 positive and 1000 negative documents. Hence, we added some additional paradigm words to the original list by reexamining a small portion of the corpus based on a very preliminary check of word frequency counts. Finally, the resulting paradigm word list contains 21 positive and 21 negative paradigm words respectively, as shown in Table 1.

Table 1: Paradigm word list.

Positive	dazzling brilliant phenomenal excellent fantastic gripping mesmerizing riveting spectacular cool awesome thrilling moving exciting love wonderful best great superb still beautiful
Negative	sucks terrible awful unwatchable hideous bad cliched boring stupid slow worst waste unexcit rubbish tedious unbearable pointless cheesy frustrated awkward disappointing

Mutual information (MI) In statistical language modeling, mutual information is a criterion widely used for calculating the semantic association between words. Here we use mutual information to select the words that have strong association with positive or negative sentiment classes. The top 20 words within each individual sentiment class were selected based on their MI scores and incorporated as prior information for our models.

Full subjectivity lexicon We also explored using the publicly available subjectivity word list with established polarities such as the MPQA subjectivity lexicon[3], which consists of 2718 positive and 4911 negative words[4]. By matching the words in the MPQA subjectivity lexicon with the vocabulary (with 25,166 distinct terms) of the movie review dataset, we finally obtained a subset of 1335 positive, 2214 negative words.

Filtered subjectivity lexicon The filtered subjectivity lexicon was obtained by removing from the full subjectivity lexicon the words occurred less than 50 times in the movie review dataset. The words whose polarity changed after stemming were also removed automatically. Finally, the filtered subjectivity lexicon contains 374 positive and 675 negative words.

Although one may argue that the paradigm word list and the MI extracted words seem requiring certain supervision information from the corpus itself, the subjectivity lexicon used here is fully domain-independent and does not bear any supervision information specifically to the movie review dataset. In fact, the JST model with the filtered subjectivity lexicon achieved better performance than the ones using the prior information obtained from paradigm word list or MI extracted words as can be seen later in Section 5. While it is well-known that sentiment classifiers trained on one domain often fail to produce satisfactory results in another domain, we speculate that the unsupervised nature of our JST model makes it highly portable to other domains.

4.3 Incorporating Prior Information

We modified Phan's GibbsLDA++ package[5] for the JST and tying-JST model implementation. In the experiments, the prior information was only utilized during the initialization of posterior distribution z, i.e. assignment of word token to sentiment label and topic. We chose a total number of 3 sentiment labels representing positive, negative and neutral, considering the fact that the sentiment of any word can be categorized into one of these three classes. The initialization starts by comparing each word token in the corpus against the words in the sentiment word list as described in Section 4.2. If there is a match, the word token is assigned with the corresponding sentiment label. Otherwise, a sentiment label is randomly sampled for a word token.

5. EXPERIMENTAL RESULTS

In this section, we will present and discuss the experimental results of both document sentiment classification and topic extraction, based on the movie review dataset.

5.1 Sentiment Classification

The document sentiment is classified based on $P(\mathbf{l}|d)$, the probability of sentiment label given document, which is approximated using Equation 8 in the implementation. In our

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

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

[4]We discarded words with 'neutral' label in the subjectivity lexicon since the number of neutral words is small and many of the neutral words have multiple polarities, e.g. both neutral and positive.

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

Table 2: Results of incorporating various prior information.

Prior information	# of polarity words (pos./neg.)	JST (%)			Tying-JST (%)		
		pos.	neg.	overall	pos.	neg.	overall
Without prior information	0/0	63	56.6	59.8	59.2	53.8	56.5
Paradigm words	21/21	70.8	77.5	74.2	74.2	71.3	73.1
Paradigm words + MI	41/41	76.6	82.3	79.5	78	73.1	75.6
Full subjectivity lexicon	1335/2214	74.1	66.7	70.4	77.6	69	73.3
Filtered subjectivity lexicon	374/675	84.2	81.5	82.8	84.6	73.1	78.9
Filtered subjectivity lexicon (subjective MR)	374/675	96.2	73	84.6	89.2	74.8	82
Pang *et al.* (2002) [16]	N/A	Classifier used: SVMs			Best accuracy: 82.9%		
Pang and Lee (2004) [15] (subjective MR)	N/A	Classifier used: SVMs			Best accuracy: 87.2%		
Whitelaw *et al.* (2005) [24]	1597 appraisal groups	Classifier used: SVMs			Best accuracy: 90.2%		
Kennedy and Inkpen (2006) [10]	1955/2398	Classifier used: SVMs			Best accuracy: 86.2%		

experiments, we only consider the probability of positive and negative label given document, with the neutral label probability being ignored. There are two main reasons. Firstly, movie review sentiment classification in our case is effectively a binary classification problem, i.e. documents are being classified either as positive or negative, without the alternative of neutral. Secondly, 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 label given document, rather than the distribution of neutral label given document. Therefore, we define that a document d is classified as a positive-sentiment document if its probability of positive sentiment label given document $P(\mathbf{l_{pos}}|d)$, is greater than its probability of negative sentiment label given document $P(\mathbf{l_{neg}}|d)$, and vice versa.

In this section, we show how prior information improves the sentiment classification accuracy of the JST and tying-JST models and how topic mixtures affect the performance of our models.

5.1.1 Results with Different Prior Information

Table 2 shows the sentiment classification accuracy at document level by incorporating various prior information. The number of polarity (positive and negative) words in various subjectivity word list is also listed. In all of the results showed in the table, α is set to $\frac{50}{\#topics}$, β is set to 0.01. It should be noted that while LDA can produce reasonable results with a simple uniform Dirichlet prior for its hyperparameters, asymmetric prior γ for sentiment-document distribution should be used since it captures different correlations among sentiment labels. In our experiments, γ is set to 0.01 for positive sentiment label and 5 for negative sentiment label. The setting for γ was determined empirically. It is worth pointing out that hyperparameters can be learned from data directly by maximum likelihood or maximum a posteriori estimation [23]. Alternatively, an approximation approach such as moment matching could also be used to avoid iterative methods for the sake of simplicity and speed [12]. We leave the estimation of γ in a more principled way as future work.

It can be observed from Table 2 that without incorporating any prior information, JST only achieved around 60% overall accuracy. By incorporating merely 21 positive and 21 negative paradigm words, a significant performance improvement is observed with JST and tying-JST giving an overall of 74.2% and 73.1% accuracy respectively. We also experimented the combination of paradigm words and mutual information and evaluated how mutual information can help to improve the sentiment classification accuracy. We extracted the top 20 positive/negative words based on the MI value calculated from the 40 randomly selected labeled documents from the movie review dataset with equal number of positive and negative documents. Plus the paradigm words listed in Table 1, the total number of positive and negative words is 41 each. It can be observed that there is a considerable improvement in classification accuracy after incorporating the MI-extracted words, with 5.3% and 2.5% improvement for JST and tying-JST respectively.

Subjectivity lexicons have attracted increasing focus in previous work [1]. Intuitively, one might expect that with a larger subjectivity lexicon and hence an increasing number of polarity words, sentiment classification performance would be improved since an overall polarity of a text can be inferred from the aggregated polarity of its individual words. However, the results shown in Table 2 reveal that incorporating the full subjectivity lexicon with 1335 positive and 2214 negative words in fact hurts the performance of both JST and tying-JST, with a relatively poor overall accuracy of 70.4% and 73.3% being achieved respectively. In contrast, with the filtered subjectivity lexicon by removing the infrequent polarity words, the performance of both models improves. Thus, the full subjectivity lexicon actually introduces more noise into the models and hence resulted in poorer performance. Also, the yielding results (82.8% for JST and 78.9% for tying-JST) are actually better than the performance by incorporating any aforementioned prior information.

We also observe that tying-JST performed consistently worse than the JST model except for the case of incorporating full subjectivity lexicon as prior information. Therefore, JST seems to be a more reasonable model design in terms of sentiment classification.

5.1.2 Results with Subjectivity Detection

In another set of experiments, we followed the approach in [15] and performed subjectivity detection (with sentences that do not express any opinions removed) prior to sentiment classification. Subjective sentences were extracted from the original movie review dataset using the LingPipe package[6]. First, we trained the subjectivity classifier based on the Sub-

[6]http://alias-i.com/lingpipe/demos/tutorial/sentiment/readme.html

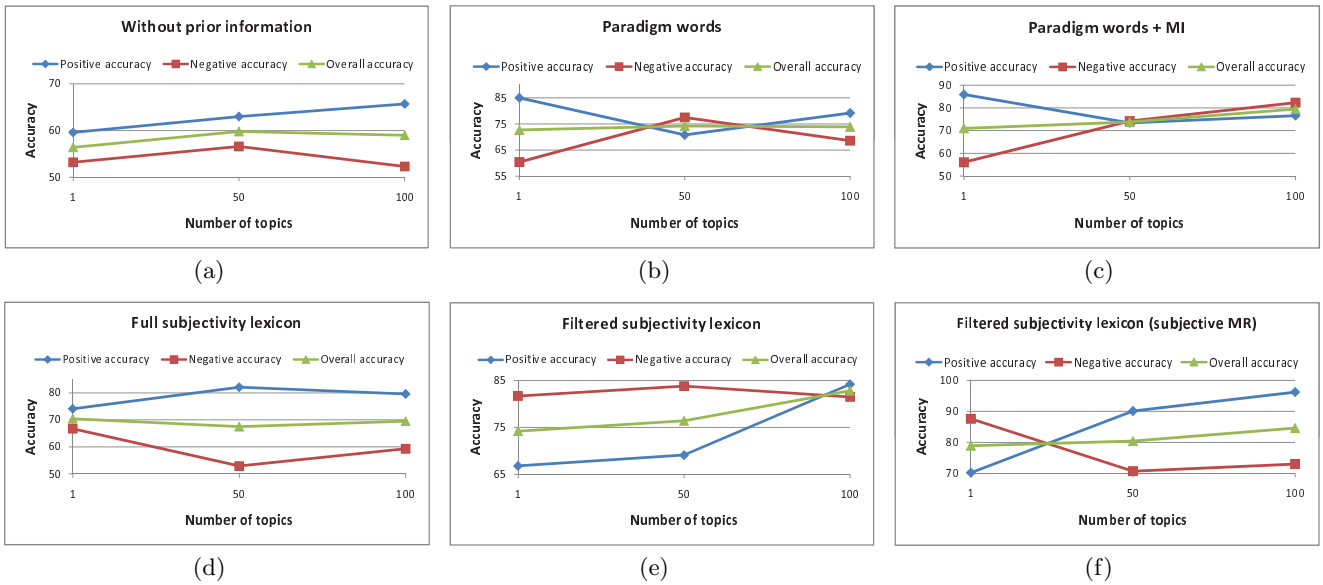

Figure 3: Sentiment classification accuracy VS. different topic numbers.

jectivity v1.0 dataset[7] which contains 5000 subjective and 5000 objective sentences. The trained classifier was then used to extract the subjective sentences from the movie review dataset, which reduces each single document to 5 to 25 sentences. After subjectivity detection and data preprocessing as described in Section 4.1, the dataset, which we named as "subjective MR", still contains 2000 documents but with a total of 334,336 words and 18,013 distinct terms (c.f. 25,166 distinct terms without subjectivity detection).

It can be seen from Table 2 that the best performance for both JST and tying-JST is obtained on the subjective MR dataset with the prior sentiment label information obtained from the filtered subjectivity lexicon, where an overall accuracy of 84.6% and 82% was achieved by JST and tying-JST respectively. This is a clear improvement over 82.8% and 78.9% when no subjectivity detection was performed. It suggests that though the subjective MR 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 [15].

5.1.3 Comparison with Existing Approaches

For comparison, document-level sentiment classification results on the movie review dataset from four previous studies are also listed in the last four rows of Table 2. The best result reported in [16] is 82.9%, which is attained by support vector machines (SVMs) using bag-of-unigram features. The performance was later further improved to 87.2% [15] by applying SVMs on the subjective portions of the movie reviews which were extracted using a subjectivity detector as described in Section 5.1.2. Whitelaw *et al.* [24] used SVMs to train on the combination of different types of appraisal group features and the bag-of-words features for sentiment analysis. The reported best accuracy is 90.2% using 1,597 appraisal groups with each possible combination of Attitude and Orientation plus 48,314 bag-of-words features. Their appraisal groups were constructed semi-automatically and comprise of a total of 41,082 appraising groups. This is much more complicated than the subjectivity lexicon used in this paper. Kennedy and Inkpen [10] combined two main sources, General Inquirer[8] and *Choose the Right Word* [7], to obtain a total of 1,955 positive and 2,398 negative terms. They then trained two classifiers, one was based on counting the number of positive and negative terms contained in movie reviews and augmented with contextual valence shifters, while the other was based on SVMs trained from the combination of unigrams and valence shifter bigrams. These two classifiers were finally combined to give the best classification accuracy which is 86.2%.

In our experiment, the best overall accuracy achieved by JST is 84.6%, based on the filtered subjectivity lexicon and the subjective MR dataset. It outperforms the best result reported in [16] and is only 2.6% and 1.6% lower than the results reported in [15] and [10]. Even for the state-of-the-art result reported in [24], the best accuracy achieved by JST is only 5.6% lower. While all the previous studies mentioned here relied on the labeled movie review data to train sentiment classifiers, our proposed JST model is fully unsupervised. In addition, the previous reported results [15, 24, 10] were all based on 10-fold cross validation in a test set comprising of 200 documents only[9], our experimental results reported here are based on the whole movie review dataset with a total of 2000 documents.

5.1.4 Results with Different Topics

We also evaluated the mixture of topics and sentiments. Figure 3 shows the sentiment classification accuracy of the JST model incorporating prior information obtained in different ways with the number of topics set to 1, 50 and 100. When the topic number is set to 1, the JST model is es-

[7]http://www.cs.cornell.edu/People/pabo/movie-review-data/

[8]http://www.wjh.harvard.edu/~inquirer/

[9][16] used an early version of the movie review data which consists of 700 positive and 700 negative documents and the results were based on 3-fold cross validation.

Table 3: Example of topics extracted by JST under different sentiment labels.

Positive sentiment label						Negative sentiment label					
Topic 1		Topic 2		Topic 3		Topic 1		Topic 2		Topic 3	
w	$P(w\|z,l)$	w	$P(w\|z,l)$	w	$P(w\|z,l)$	w	$P(w\|z,l)$	w	$P(w\|z,l)$	w	$P(w\|z,l)$
good	0.084708	tom	0.035175	ship	0.059020	bad	0.079132	sex	0.065904	prison	0.073208
realli	0.046559	ryan	0.030281	titan	0.031586	worst	0.035402	scene	0.053660	evil	0.032196
plai	0.044174	hank	0.025388	crew	0.024439	plot	0.033687	sexual	0.031693	guard	0.031755
great	0.036645	comedi	0.021718	cameron	0.024439	stupid	0.029767	women	0.026291	green	0.029109
just	0.028990	star	0.020800	alien	0.022826	act	0.025602	rate	0.023770	hank	0.028227
perform	0.028362	drama	0.016519	jack	0.020751	suppos	0.025480	act	0.023230	wonder	0.027345
nice	0.026354	meg	0.015601	water	0.019137	script	0.024500	offens	0.018728	excute	0.026904
fun	0.025978	joe	0.014378	stori	0.017984	wast	0.024500	credict	0.016027	secret	0.025581
lot	0.025853	relationship	0.014072	rise	0.016601	dialogu	0.023643	porn	0.014587	mile	0.022936
act	0.022715	mail	0.013766	rose	0.013835	bore	0.022908	rape	0.013867	death	0.022495
direct	0.021586	blond	0.013460	boat	0.013374	poor	0.022908	femal	0.013686	base	0.022054
best	0.020331	run	0.012543	deep	0.013143	complet	0.020825	cut	0.013686	tom	0.019849
get	0.020331	phone	0.012237	ocean	0.012451	line	0.019968	gril	0.013506	convict	0.018967
entertain	0.018198	date	0.011931	board	0.011990	terribl	0.018988	parti	0.012426	return	0.018526
better	0.017445	got	0.011625	sink	0.011299	mess	0.015313	male	0.011886	franklin	0.016762
job	0.016692	busi	0.011319	sea	0.010838	wors	0.014333	bad	0.011346	happen	0.016321
talent	0.016064	cute	0.011013	rain	0.010838	dull	0.013598	nuditi	0.011166	power	0.014116
pretti	0.016064	sister	0.010708	dicaprio	0.010607	actor	0.012986	woman	0.010986	known	0.012352
try	0.015688	children	0.010096	storm	0.010377	total	0.012986	peopl	0.010986	instinct	0.011470
want	0.015186	dog	0.009790	disast	0.010146	isn	0.012863	nake	0.010625	inmat	0.011470

sentially transformed to a simple LDA model with only S topics, each of which corresponds to a sentiment label. Consequently, it ignores the correlation between sentiment labels and topics. It can be observed from Figure 3 that, JST performs worse with single topic compared to 50 and 100 topics, except for the case of full subjectivity lexicon as shown in Figure 3(d) where the single topic performance is almost the same as the one with 100 topics. For paradigm words + MI, filtered subjectivity lexicon and filter subjectivity lexicon (subjective MR) (Figures 3(c), 3(e), and 3(f)), the result with 100 topics outperforms the ones with other topic number settings. For the case when no prior information is applied as well as paradigm words as shown in Figure 3(a) and Figure 3(b), the results with 50 topics are almost the same as the ones achieved with 100 topics and both are higher than that of the single topic setting. It can be also easily seen that the results with filtered subjectivity lexicon in Figure 3(e) give the most balanced classification accuracy on both positive and negative documents. From the above, we can conclude that topic information indeed helps in sentiment classification as the JST model with the mixture of topics consistently outperforms a simple LDA model ignoring the mixture of topics. This justifies the proposal of our JST model. Also, the empirical results reveal that the optimum number of topics for the movie review dataset is 100.

5.2 Topic Extraction

The second goal of JST is to extract topics from the movie review dataset (without subjectivity detection) and evaluate the effectiveness of topic sentiment captured by the model. In the experiment, the distribution of words given topic and sentiment label was estimated using Equation (6). Unlike the LDA model that a word is drawn from the topic-word distribution, in JST one draws a word from the distribution over words conditioned on both topics and sentiment labels. Therefore, we analyze the extracted topics under two different sentiment labels (positive and negative). Six example topics extracted from the movie review dataset under positive and negative sentiment labels are shown in Table 3.

The three topics on the left columns of Table 3 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 20 topic words. As can be seen from the table that the six extracted topics are quite informative and coherent, where each of them tried to capture the underlying theme of a movie or the relevant comments from a movie reviewer. For example, under the positive sentiment label category, topic 1 is likely to be very positive review comments for a movie; topic 2 is apparently about the movie *"You've got a mail"* by Tom Hanks and Meg Ryan; topic 3 is closely related to the very popular romantic movie *"Titanic"* directed by James Cameron and casted by Leonardo DiCaprio and Kate Winslet. For the topics under the negative sentiment category, topic 1 is probably the criticism made by a movie reviewer, while topic 2 is about movies related to sex/porn issues and topic 3 is likely to be the movie *"Green Mile"* by Tom Hanks.

In terms of topic sentiment, by examining each of the topics in Table 3, it is quite evident that topic 1 under the positive sentiment label and topic 1 under the negative label indeed bear positive and negative sentiment respectively. For topic 2 and topic 3 under the negative sentiment label, it is still fairly easy to recognize that some of their topic words convey negative sentiments though not as strong as the ones in topic 1. Topic 2 and topic 3 under the positive sentiment label mainly describe movie plots with less words carrying positive sentiment compared to topic 1 under the same category. Manually examining the data reveals that the terms that seem not conveying sentiments under these two topics in fact appear in the context expressing positive sentiments. The above analysis illustrates the effectiveness of JST in extracting mixture of topics from a corpus.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a joint sentiment/topic (JST) model which can detect document level sentiment and extract mixture of topics from text simultaneously. In contrast to most of the existing approaches in sentiment classification which rely on supervised learning, the proposed JST model is fully unsupervised, thus provides more flexibilities and can be easier adapted to other applications. Experiments have been conducted to evaluate the performance of JST based on the movie review dataset. The preliminary results demonstrated that our model is able to give competitive performance in document level sentiment classification compared with the results generated by other existing supervised approaches and the discovered topics are indeed coherent and informative.

One of the limitations of our model is that it represents each document as a bag of words and thus ignores the word ordering. It will probably predict the sentiment of "not good movie" being positive and the sentiment of "not bad movie" being negative. Thus, in future work, we will extend the model to include higher order information (bigrams or trigrams). Another promising future step is to extend JST to detect the polarity of text at various granularity levels, e.g. detecting sentiment labels for more fine-grained topics. We also intend to carry out a large scale of experiments and evaluate the model performance on datasets from different domains.

7. ACKNOWLEDGMENTS

We thank Rui Wang and Naihui He for processing part of the prior information and Lei Wang for providing computing resources.

8. REFERENCES

[1] A. Abbasi, H. Chen, and A. Salem. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3):1–34, 2008.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[3] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[4] Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[5] K. Eguchi and V. Lavrenko. Sentiment retrieval using generative models. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 345–354, Sydney, Australia, July 2006. Association for Computational Linguistics.

[6] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(90001):5228–5235, 2004.

[7] S. Hayakawa and E. Ehrlich. *Choose the right word: A contemporary guide to selecting the precise word for every situation.* Collins, 1994.

[8] T. Hofmann. Probabilistic latent semantic indexing. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, New York, NY, USA, 1999. ACM.

[9] N. Kaji and M. Kitsuregawa. Automatic construction of polarity-tagged corpus from html documents. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 452–459, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

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

[11] S.-M. Kim and E. Hovy. Determining the sentiment of opinions. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 1367, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[12] W. Li and A. McCallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 577–584, 2006.

[13] R. McDonald, K. Hannan, T. Neylon, M. Wells, and J. Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 432–439, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[14] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 171–180, New York, NY, USA, 2007. ACM.

[15] B. Pang and L. Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 271, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[16] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 79–86, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[17] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[18] M. Steyvers and T. Griffiths. Probabilistic Topic Models. *Handbook of Latent Semantic Analysis*, page 427, 2007.

[19] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[20] I. Titov and R. McDonald. Modeling online reviews

with multi-grain topic models. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 111–120, New York, NY, USA, 2008. ACM.

[21] P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, Morristown, NJ, USA, 2001. Association for Computational Linguistics.

[22] P. D. Turney and M. L. Littman. Unsupervised learning of semantic orientation from a hundred-billion-word corpus. *CoRR*, cs.LG/0212012, 2002.

[23] H. M. Wallach. Topic modeling: beyond bag-of-words. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.

[24] C. Whitelaw, N. Garg, and S. Argamon. Using appraisal groups for sentiment analysis. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631, New York, NY, USA, 2005. ACM.

[25] J. Zhao, K. Liu, and G. Wang. Adding redundant features for CRFs-based sentence sentiment classification. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 117–126, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.