Twitter Opinion Topic Model: Extracting Product Opinions from Tweets by Leveraging Hashtags and Sentiment Lexicon

Kar Wai Lim
ANU & NICTA, Canberra, Australia
karwai.lim@anu.edu.au

Wray Buntine
Monash University, Melbourne, Australia
wray.buntine@monash.edu

ABSTRACT

Aspect-based opinion mining is widely applied to review data to aggregate or summarize opinions of a product, and the current state-of-the-art is achieved with Latent Dirichlet Allocation (LDA)-based model. Although social media data like tweets are laden with opinions, their "dirty" nature (as natural language) has discouraged researchers from applying LDA-based opinion model for product review mining. Tweets are often informal, unstructured and lacking labeled data such as categories and ratings, making it challenging for product opinion mining. In this paper, we propose an LDA-based opinion model named Twitter Opinion Topic Model (TOTM) for opinion mining and sentiment analysis. TOTM leverages *hashtags*, *mentions*, emoticons and strong sentiment words that are present in tweets in its discovery process. It improves opinion prediction by modeling the target-opinion interaction directly, thus discovering target specific opinion words, neglected in existing approaches. Moreover, we propose a new formulation of incorporating sentiment prior information into a topic model, by utilizing an existing public sentiment lexicon. This is novel in that it learns and updates with the data. We conduct experiments on 9 million tweets on electronic products, and demonstrate the improved performance of TOTM in both quantitative evaluations and qualitative analysis. We show that aspect-based opinion analysis on massive volume of tweets provides useful opinions on products.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: NLP—*Text analysis*

General Terms

Design, Experimentation

Keywords

Opinion mining, sentiment analysis, Twitter, topic modeling, product review, sentiment lexicon, emoticons

1. INTRODUCTION

When making a purchase decision, a key deciding factor can often be the reviews written by other consumers. These reviews are freely available online, however, one can rarely read all the reviews given their volume. This has led to various automated algorithms to mine the reviews, extracting a more digestible summary for a user. The task of analyzing opinions from text data such as reviews is known as opinion mining or opinion extraction [19, 33].

Among various approaches to opinion mining, *aspect-based opinion mining* has recently gained a lot of attention from the research community. Aspect-based opinion mining involves extracting the major aspects or facets from data for analysis. As an example, for a camera product, the aspects could be "picture quality", "portability" *etc*. Topic models are often used to determine the aspects through soft clustering. Topic models have been successfully applied to review data crawled from review websites such as Epinions.com, TripAdvisor *etc*. LDA-based models are considered to be state-of-the-art for aspect-based opinion mining [28].

Besides reviews extracted from review websites, opinions from social media websites are also very useful, even though they are often overlooked as a source for reviews. Social media text is short and is regarded as "dirty", and hence less useful for more sophisticated language analysis [47]. The same problem also leads to degradation when applying NLP tools [36]. Despite these limitations, large numbers of tweets containing opinions are generated every day and are very relevant for opinion mining. We argue that while tweets are generally unstructured, Twitter is a useful source of reviews since it provides a convenient platform for users to express their opinions. Twitter is also integrated to a person's social life, making it easier for users to express their opinions on products by tweeting instead of writing a review on review websites.

In this paper, we demonstrate the usefulness of Twitter as a source for aspect-based target-opinion mining. We propose a novel LDA-based opinion model that is designed for tweets, which we name Twitter Opinion Topic Model (TOTM). TOTM models the target-opinion interaction directly, which significantly improves opinion prediction, *e.g.* TOTM discovers *'grilled'* is positive for *sausage* but not other targets. We note that while there are no explicit ratings and scores on tweets, tweets often contain emoticons and strong sentiment words, such as '*love*' and '*hate*'. TOTM exploits this fact and uses the information to compensate for the lack of explicit ratings. Additionally, *hashtags* are strong indicators of topics for tweets [24]. TOTM makes use of the hashtags and *mentions* in tweets for tweet aggregation, which improves aspect clustering. Modeling with TOTM also allows us to acquire additional summaries on products, which are not obtainable with existing models.

Furthermore, we incorporate a sentiment lexicon as prior information into TOTM. We propose a novel formulation of how the sentiment lexicon affects the priors in TOTM. Our approach facilitates automatic learning of the lexicon strength based on the data; while current existing methods are *ad hoc* or ruled-based. Our for-

CIKM'14, November 3–7, 2014, Shanghai, China.

ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2662005.

mulation is shown to perform best for sentiment classification. Additionally, we propose a different target-opinion extraction procedure that works better for tweets, discussed in Subsection 8.1. We note that text preprocessing is important when dealing with tweets.

We apply TOTM to 3 tweets corpus, showing improved performance of TOTM in model fitting and sentiment analysis. Qualitatively, we demonstrate the usefulness of TOTM in extracting the opinions on products from tweets. As large volumes of tweets laden with opinions are generated daily, real-time aspect-based opinion analysis allows us to obtain first-hand opinions on new products, which might not be as readily available from review websites.

The rest of the paper is structured as follows. Section 2 reviews some related work, and Section 3 provides a summary of our task and major contributions. In Section 4, we present Interdependent LDA (ILDA) [27] which will be used as a baseline for comparison. We introduce TOTM in Section 5 and the method of incorporating a lexicon in Section 6. In Section 7, we discuss TOTM's model likelihood and inference procedure, as well as proposing a novel hyperparameter sampling procedure. We then describe the data used in this paper and report on the experiments in Sections 8 and 9. Finally, we conclude the paper in Section 10.

2. RELATED WORK

Latent Dirichlet Allocation (LDA) is a topic model that has been extended by many for sentiment analysis. Notable examples based on LDA include the MaxEnt-LDA hybrid model [48], Joint Sentiment Topic (JST) model [18], Multi-grain LDA (MG-LDA) [43], Interdependent LDA (ILDA) [27], Aspect and Sentiment Unification Model (ASUM) [15] and Multi-Aspect Sentiment (MAS) model [42]. The Topic-Sentiment Mixture (TSM) model [25] performs sentiment analysis by utilizing the Multinomial distribution. These models perform aspect-based opinion analysis and they had been successfully applied to review data of different domains, such as electronic product, hotel and restaurant reviews. The task of summarizing the reviews is also known as *opinion aggregation*.

To the best of our knowledge, there is no existing LDA-based opinion aggregation method that has been successfully applied to social media data such as tweets. Current opinion mining methods that are used on tweets tend to be *ad hoc* or rule-based. We suspect this is because tweets are generally regarded as too noisy for model-based methods to work, and also due to the fact that LDA works badly on short documents. Maynard et al. [22] studied the challenges in developing an opinion mining tool for social media and they advocated the use of shallow techniques in linguistic processing of tweets. Notable non-LDA-based methods for opinion analysis include OPINE [35], which uses relaxation labeling to classify sentiment, and Opinion Digger [26], an aspect-based review miner using *k nearest neighbor*. Hu and Liu [12] performed rule-based target-opinion extraction from online product reviews, while Li et al. [16] extracted opinions from reviews using Conditional Random Fields. On tweets, Pak and Paroubek [32] performed opinion analysis using a Naive Bayes classifier; while Liu et al. [20] performed sentiment classification using an adaptive co-training SVM. Go et al. [8] and Davidov et al. [4] made use of emoticons (smileys), which were found to provide improvement for sentiment classification on tweets. Since tweets are always short, existing work [8, 32, 4, 20] tends to assume a single polarity for each tweet. In contrast, Jiang et al. [14] performed target-dependent sentiment analysis, where the sentiments apply to a specific target.

Lexical information can be used to improve sentiment analysis. He [11] used a sentiment lexicon to modify the priors of LDA for sentiment classification, though with an approach with *ad hoc* constants. Li et al. [17] incorporated a lexical dictionary into a non-negative matrix tri-factorization model, using a simple rule-based polarity assignment. Refer to Ding et al. [6] and Taboada et al. [37] for a detailed review on applying lexicon-based methods in sentiment analysis. Instead of a lexicon, Jagarlamudi et al. [13] used seeded words as lexical priors for semi-supervised topic modeling.

3. OPINION MINING TASK ON TWEETS

In this section, we describe the opinion mining problem we are tackling and outline our major contributions in solving the problem.

3.1 Problem Definition

Given a collection of documents (tweets), our first problem is to extract *target-opinion* pairs from each document. A target-opinion pair $\langle t, o\rangle$ consists of two phrases: a *target* phrase t which is the object being described, and an *opinion* phrase o which is the description. Target phrases are usually nouns and opinion phrases are usually adjectives, examples include $\langle$*picture quality, good*$\rangle$, $\langle$*iPhone app, expensive*$\rangle$ *etc*. Note that a phrase can be either a collocation (multi-word phrase) or a single word. For simplicity, we will use '*word*' to mean a *single-word* or a *phrase* in this paper.

Our next problem is to group the target-opinion pairs into clusters and identify the associated sentiments. The produced clusters should depend on the tweet corpus, as they should represent different aspects of the corpus. For example, given a tweet corpus which consists of various electronic products, we would like different products to be grouped into different clusters. Each target-opinion pair is assigned 2 latent labels, the first being *aspect* a indicating which cluster the pair belongs, the second label being *sentiment* r. The sentiment of a target-opinion pair refers to the polarity of the opinion phrase, which can be *negative*, *neutral* or *positive*.

Finally, we would like to display a summary (high level view) of the obtained quadruples $\langle t, o, a, r\rangle$. There are many ways to do this, here we follow the standard topic modeling approach to display the top phrases. We inspect the target phrases given the aspects. We also examine the opinion phrases given the target phrases and sentiments. In brief, our task of opinion mining on tweets is to extract useful opinions and represent them in a format that is easy to digest. For example, with a tweet corpus on electronic products, we would like to discover the opinions of Twitter users on certain products, such as iPhones.

3.2 Major Contributions

We make two major contributions as follows: Firstly, *we design an LDA-based topic model* (TOTM) for performing aspect-based target-opinion analysis on product reviews from tweets. TOTM is novel in that it directly models the target-opinion interaction, giving significant improvement in opinion prediction. Existing aspect-based methods only model the interaction between aspects and sentiments, leaving the targets and opinions to be weakly associated through aspects and sentiments. Without this explicit modeling, the existing models failed to sensibly assign opinions to targets. For example, from a restaurant review with *friendly staff* and *delicious cake*, existing LDA-based opinion model failed to recognize that *friendly* cannot be used to describe *cake*. Also, as mentioned in the introduction, TOTM makes use of available auxiliary variables in tweets (hashtags, mentions, emoticons and strong sentiment words) to improve aspect-based opinion analysis.

Secondly, *we propose a new formulation for incorporating a sentiment lexicon* into our topic model. While existing methods adopt an *ad hoc* or ruled-based approach to incorporating sentiment prior, our formulation is novel in that it is learned automatically given the data. This is done robustly using a tuning hyperparameter that is optimized automatically. The sentiment information is used to adjust the opinion priors in order to improve sentiment analysis.

4. BASELINE: INTERDEPENDENT LDA

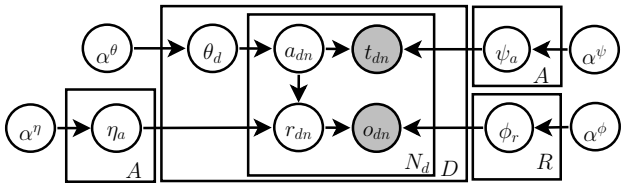


Figure 1: Graphical Model for Interdependent LDA

Interdependent LDA (ILDA) [27] is an extension of LDA that performs aspect-based opinion analysis. It jointly models the aspect (a) and sentiment[1] (r) for each target-opinion pair $\langle t, o\rangle$ that is present in a document. We note that the sentiment variable r is a categorical variable, and is not restricted to just 3 values. However, in this paper, we will assume that the sentiment r has only three labels $\{-1, 0, 1\}$, which correspond to negative, neutral and positive sentiment respectively.

In this paper, we treat ILDA as a baseline. It has the following generative process. For each document d, we sample a document-aspect distribution

$$\theta_d \sim \text{Dir}(\alpha^\theta) \ .$$

For each aspect a, we sample an aspect-sentiment distribution η_a and an aspect-target word distribution ψ_a:

$$\eta_a \sim \text{Dir}(\alpha^\eta) \ , \qquad \psi_a \sim \text{Dir}(\alpha^\psi) \ .$$

Given each sentiment r, we sample a sentiment-opinion phrase distribution

$$\phi_r \sim \text{Dir}(\alpha^\phi) \ .$$

Finally, we model each target-opinion pair $\langle t_{dn}, o_{dn}\rangle$ and their respective latent aspects and sentiments.

$$a_{dn} \sim \text{Discrete}(\theta_d) \ , \qquad r_{dn} \sim \text{Discrete}(\eta_{a_{dn}}) \ ,$$
$$t_{dn} \sim \text{Discrete}(\psi_{a_{dn}}) \ , \qquad o_{dn} \sim \text{Discrete}(\phi_{r_{dn}}) \ .$$

We note that the α's are the hyperparameters corresponding to the symmetric Dirichlet distributions.

ILDA models the sentiment conditionally on the aspect; and given the aspect and sentiment, the target word and opinion word are generated independently. Although such modeling is often adequate (since many of the opinion words can be applied generally to most target words), it fails to take into account that some opinion words are restricted to certain target words, and *vice versa*. For example, we can say a phone has *short battery life* but not *short camera quality*.

In this paper, we do not compare against other models such as MG-LDA and ASUM, since these models do not perform target-based opinion analysis, and thus not directly comparable.

5. TWITTER OPINION TOPIC MODEL

Here we present the Twitter Opinion Topic Model for aspect-based opinion analysis on tweets. The model is given in Figure 2. Contrary to ILDA, we do not model the aspect-sentiment distribution η. Instead, we model the target-opinion pairs directly. This allows us to better model the opinion words, and also provides us with a finer level of opinion analysis. For example, TOTM will be able to model that the word *'limited'* can describe *battery life* but is unlikely to be used to describe *charger*.

[1]Also known as *rating* in Moghaddam and Ester [27].

Table 1: List of Variables for TOTM

Variable	Description
a	Aspect: category label for a target-opinion pair; also known as topic in topic models.
r	Sentiment: polarity of an opinion phrase.
t	Target: word or phrase that is being described.
o	Opinion: description of a target word t.
e	Emotion Indicator: binary variable indicating positive or negative emotion; can be unobserved.
ψ	Target word distribution: Probability distribution for target words.
ϕ, ϕ', ϕ^*	Opinion word distribution: Probability distribution for opinion words.
γ	Sentiment distribution: Probability distribution in generating a sentiment label r.
α, β	Hyperparameters associated with the PYP.
H	Base distribution for the PYP.

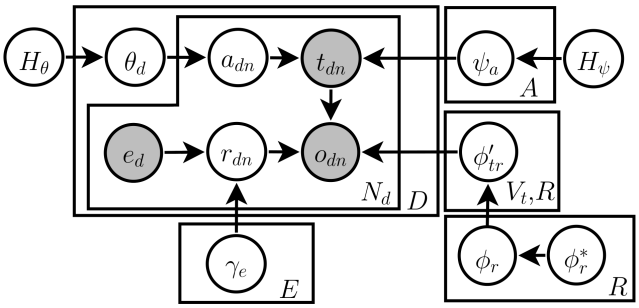


Figure 2: Graphical Model for Twitter Opinion Topic Model

TOTM uses the *Griffiths-Engen-McCloskey* (GEM) [34] distribution to generate probability vectors and the *Pitman-Yor process* (PYP) [39] to generate probability vector given another mean probability vector. Both GEM and PYP are parameterized by a discount parameter α and a concentration parameter β; and PYP is additionally parameterized by a mean or base distribution H. The GEM distribution is equivalent to the PYP with a base distribution that generates an ordered integer label, H_θ. The PYP is also known as the two-parameter Poisson-Dirichlet process.

We introduce a variable e named *emotion indicator*, which detects the existence of emoticons and/or strong sentiment words in the documents. The strong sentiment words are hand-selected and represent words that are associated with a person's positive or negative feeling. We present some examples of strong sentiment words in Table 2, and provide the full list in the supplementary material made available online on the author's website. We define e to be -1 when only a negative emotion is observed and e to be 1 when only a positive emotion is observed, otherwise we treat e as unobserved. Note that $e = 0$ would correspond to a neutral emotion, but we have no such observations so this is not considered.

The generative process of TOTM is as follows. First, we sample the document-aspect distribution θ_d for each document d,

$$\theta_d \sim \text{GEM}(\alpha^\theta, \beta^\theta) \ .$$

Second, for $e = \{-1, 1\}$, we model the emotion-sentiment distribution γ_e by a Dirichlet distribution with asymmetric prior:

$$\gamma_e | e \sim \text{Dir}(\vec{q}_e) \ .$$

The prior q_e is chosen such that $\vec{q}_{-1} = (0.9, 0.05, 0.05)$ and $\vec{q}_1 = (0.05, 0.05, 0.9)$.

Next, for the target words, we generate the aspect-target distribution ψ_a for each aspect a:

$$\psi_a \sim \text{PYP}(\alpha^{\psi}, \beta^{\psi}, H_{\psi}) \ .$$

Here, H_{ψ} is a discrete uniform vector over the vocabulary of the target words (V_t).

For the opinion words, we propose a novel hierarchical modeling that allows an opinion word to describe two different targets differently (*e.g. short* for *processing time* is good but *short* for *battery life* is bad), while at the same time allows for sharing of the polarity of opinion words between targets. This is achieved by assigning common base distributions to the target-opinion distributions. So target-opinion distributions ϕ'_{tr} for different targets t share a common mean ϕ_r which itself is unknown so we sample it from a uniform base ϕ^*_r. More specifically, for each $r = \{-1, 0, 1\}$ and $t = \{1, \ldots, |V_t|\}$, we generate ϕ'_{tr} as follows:

$$\phi^*_r = 1/|\vec{V}_o| \ ,$$
$$\phi_r|\phi^*_r \sim \text{PYP}(\alpha^{\phi}, \beta^{\phi}, \phi^*_r) \ ,$$
$$\phi'_{tr}|\phi_r \sim \text{PYP}(\alpha^{\phi'}, \beta^{\phi'}, \phi_r) \ ,$$

where V_o is the vocabulary of the opinion words.

Finally, for each target-opinion pair $\langle t_{dn}, o_{dn} \rangle$ (indexed by n) in document d, we sample the respective aspect a_{dn}, sentiment r_{dn} and the target-opinion pair:

$$a_{dn}|\theta_d \sim \text{Discrete}(\theta_d) \ ,$$
$$r_{dn}|e_d, \gamma \sim \text{Discrete}(\gamma_{e_d}) \ ,$$
$$t_{dn}|a_{dn}, \psi \sim \text{Discrete}(\psi_{a_{dn}}) \ ,$$
$$o_{dn}|t_{dn}, r_{dn}, \phi' \sim \text{Discrete}(\phi'_{t_{dn}, r_{dn}}) \ .$$

We note that each PYP distribution is parameterized by its own set of hyperparameters, *i.e.* β^{θ} differs for different document d, albeit not explicitly shown above for readability. We present a list of variables associated with TOTM in Table 1. Also note that by modeling the target-opinion distribution explicitly, we have to store the information of the distribution for each target in the data, which is very large. In our implementation, we adopt a sparse representation for storing the counts associated with the target-opinion distributions. We find that each target word is only described by a limited number of opinion words in the data, which is less than 1% of the words from the opinion word vocabulary.

In the next section, we propose a novel method to incorporate sentiment prior information for opinion analysis.

6. INCORPORATING SENTIMENT PRIOR

He [11] proposed a simple yet effective way to incorporate sentiment prior information into LDA by directly modifying the Dirichlet prior based on available sentiment lexicons. Naming her model LDA-DP (LDA with Dirichlet Prior modified), He replaces the topics in LDA by latent sentiment labels and allows the word priors to be custom probability distributions. The generative process of LDA-DP is identical to LDA and hence omitted in this paper.

In LDA-DP, the word distribution ϕ_r is Dirichlet distributed with the parameter $(\vec{\lambda}_r \times \alpha_r)$, where $r = \{-1, 0, 1\}$ is the sentiment label corresponding to negative, neutral and positive sentiment, respectively[2]. The λ_{rv} is initialized to be $1/3$, and subsequently updated if the sentiment lexicon contains word v. In this case, λ_{rv} takes the value of 0.9 if the sentiment of word v matches r, and takes the value of 0.05 otherwise:

$$\lambda_{rv} = \begin{cases} 0.9 & \text{if Sentiment}(v) = r \\ 0.05 & \text{otherwise} \end{cases}$$

Motivated by this, but not wishing to be required to give the exact strength by which the dictionary affects probabilities, instead, we propose a novel formulation that automatically learns and updates itself. We assume that a sentiment lexicon is available and provides sentiment scores for opinion words. Additionally, we assume that the sentiment score S_v returned from the sentiment lexicon takes negative value when v has negative sentiment, positive value when v has positive sentiment, and 0 when v is neutral[3].

Sentiment lexicons that are freely available online include SentiWordNet [1], SentiStrength [41], MPQA Subjectivity lexicon [45] and others. SentiStrength is developed from MySpace[4] text data by a research group (Statistical Cybermetrics Research Group) from the University of Wolverhampton, UK. Since the SentiStrength lexicon is constructed for informal text, we use it to extract sentiment information for TOTM. The sentiment score S_v from SentiStrenth ranges from -5 to $+5$, which conforms to our assumption. We assume that $S_v = 0$ for unlisted words.

Additionally, we make use of the SentiWordNet 3.0 lexicon to evaluate TOTM. SentiWordNet is built on WordNet [7] by researchers from Italy. We note that SentiStrength and SentiWordNet are developed independently by different teams using different methods. Thus we claim it is fair and unbiased to use one lexicon for training and the other for evaluation.

Our formulation is as follows, introducing a tunable parameter b that controls the strength of the prior, we replace the prior ϕ^*_r (in the context of TOTM) by the following:

$$\phi^*_{rv} \propto (1 + b)^{X_{rv}} \ , \tag{1}$$

where $b > 0$ and hence $\phi^*_{rv} > 0$. Here, X_{rv} is the score of word v for sentiment r, which is defined as

$$X_{rv} = \begin{cases} S_v & \text{if } r = 1 \text{ (positive)} \\ -|S_v| & \text{if } r = 0 \text{ (neutral)} \\ -S_v & \text{if } r = -1 \text{ (negative)} \end{cases} .$$

Note that although there are multiple ways to formulate the prior, we choose the above formulation due to its simplicity and intuitiveness. We can see that positive X_{rv} boosts the probability of word v while a negative X_{rv} diminishes it. Also, this formulation ensures the positivity of the prior, which can be difficult to achieve if we use other formulations such as a polynomial function.

Even though b is a tunable parameter, we do not need to manually tune it. We propose a flexible way to learn the parameter b from its posterior distribution (detailed in Subsection 7.2), thus relieving us from choosing the value for b, which can be difficult (the value of b should depend on the sentiment score of the lexicon).

7. INFERENCE TECHNIQUE

In this section, we discuss the collapsed Gibbs sampler for TOTM, and then discuss the sampling of the hyperparameters.

7.1 Collapsed Gibbs Sampling for TOTM

The key to Gibbs sampling with PYPs is to marginalize out the probability vectors (*e.g.* θ) in the model and record various associated counts instead, thus yielding a collapsed sampler. While a

[2]We redefined the original sentiment labels [11] for consistency.

[3]We can simply normalize the score to conform to this assumption.

[4]MySpace is a social networking website similar to Facebook.

Algorithm 1 Collapsed Gibbs Sampling for TOTM

1. Initialize the model by assigning a random aspect to each target-opinion pair, sampling the sentiment label, and building the relevant customer counts $c_k^{\mathcal{N}}$ and table counts $c'^{\mathcal{N}}_k$ for all nodes.
2. For each document d:
   (a) For each target phrase t_{dn}:
      i. Decrement counts associated with t_{dn}.
      ii. Sample new aspect a_{dn} and corresponding parts of $\mathbf{C}$ from Equation 4.
      iii. Increment associated counts for the new a_{dn}.
   (b) For each opinion phrase o_{dn}:
      i. Decrement counts associated with o_{dn}.
      ii. Sample new sentiment r_{dn} and corresponding parts of $\mathbf{C}$ (like Equation 4).
      iii. Increment associated counts for the new r_{dn}.
3. Repeat step 2 until the model converges or when a fixed number of iterations is reached.

common approach here is to use the Chinese Restaurant Process (CRP) representation of Teh and Jordan [40], we use another representation that requires no dynamic memory and has better inference efficiency [3]. We let $g(\mathcal{N})$ be the marginalized likelihood associated with the probability vector $\mathcal{N}$. The vector is marginalized out, thus the likelihood is in terms of — using the CRP terminology — the *customer counts* $\mathbf{c}^{\mathcal{N}} = (\dots, c_i^{\mathcal{N}}, \dots)$ and the total customer count $C^{\mathcal{N}}$ (the sum of $c_i^{\mathcal{N}}$). For the PYP, we introduce the *table counts* $\mathbf{c}'^{\mathcal{N}} = (\dots, c_i'^{\mathcal{N}}, \dots)$ that represents the subset of $\mathbf{c}^{\mathcal{N}}$ that gets passed up the hierarchy (as customer for the parent probability vector of $\mathcal{N}$), and $C'^{\mathcal{N}}$, the total table count. For instance, looking at the sub-hierarchy in Figure 2 for $\phi'_{tr} \leftarrow \phi_r \leftarrow \phi^*_r$, the customer count $c_v^{\phi'_{tr}}$ for opinion index v is associated with the table count $c'^{\phi'_{tr}}_v$ which are added to the customer count $c_v^{\phi_r}$ (ϕ_r is the parent of ϕ'_{tr}). The table count of ϕ_r, $c'^{\phi_r}_v$, is in turn added to the customer count $c_v^{\phi^*_r}$. Note that table count is always smaller than customer count ($c_i'^{\mathcal{N}} \leq c_i^{\mathcal{N}}$). These counts are latent, not observed, hence they are sampled during inference.

By using the above representation, we do not need to record the occupancy counts of each table, hence we do not need a dynamic storage. The marginalized likelihood is given by

$$g(\mathcal{N}) = \frac{(\beta^{\mathcal{N}}|\alpha^{\mathcal{N}})_{C'^{\mathcal{N}}}}{(\beta^{\mathcal{N}})_{C^{\mathcal{N}}}} \prod_i S^{c_i^{\mathcal{N}}}_{c_i'^{\mathcal{N}},\alpha^{\mathcal{N}}} \quad , \tag{2}$$

where $S^x_{y,\alpha}$ is the generalized Stirling number, whereas $(x)_C$ and $(x|y)_C$ denote the Pochhammer symbol [2].

We use bold face capital letters to denote the set of all relevant lower case variables, *e.g.* $\mathbf{A} = \{\vec{a}_1, \cdots, \vec{a}_D\}$, where each $\vec{a}_i = \{a_{i1}, \cdots, a_{i,N_d}\}$, denotes the set of all aspects. Variables $\mathbf{R}$, $\mathbf{T}$ and $\mathbf{O}$ are defined similarly. In addition, we denote $\mathbf{C}$ to be the set of the customer counts and table counts for all probability vectors ($c^{\phi'_{tr}}, c^{\phi_r}, c^{\phi^*_r}$, *etc.*) Also, we denote ζ the set of all hyperparameters (such as the α's). Note all probability vectors are marginalized out. The likelihood of the model can then be written — in terms of $g(\cdot)$ — as $p(\mathbf{A}, \mathbf{R}, \mathbf{T}, \mathbf{O}, \mathbf{C}|\zeta) \propto$

$$\Big(\prod_{d=1}^{D} g(\theta_d)\Big)\Big(\prod_{e=\{-1,1\}} g(\gamma_e)\Big)\Big(\prod_{a=1}^{A} g(\psi_a)\Big)\Big(\prod_{r=-1}^{1} g(\phi_r)\Big(\prod_{t=1}^{|V_t|} g(\phi'_{tr})\Big)\Big). \tag{3}$$

We use the collapsed Gibbs sampler from Chen et al. [3] for inference. The concept of the sampler is analogous to LDA, which consists of decrementing counts associated with a word, sampling the respective new latent values for the word, and incrementing the respective counts. In our case, the process is more complicated, albeit following the same general procedure. For the decrementing procedure, the table counts are represented as a sum of Bernoulli "indicator" variables u. Each data item (customer) corresponding to a $+1$ in $c_i^{\mathcal{N}}$ either has $u = 0$ or $u = 1$. When $u = 1$, the data item is passed up the hierarchy to the parent of $\mathcal{N}$, and thus contributes a $+1$ to the table count $c'^{\mathcal{N}}_i$. Note that the counts can only increase or decrease by one, since we are decrementing and incrementing a word at a time.

When sampling a new aspect a or sentiment r, the modularized likelihood (Equation 3) allows the posterior to be computed quickly, since the conditional posterior simplifies to a ratio of likelihoods. This in turn allows for the ratio to simplify further since the counts can only change by 1. For instance, the ratio of the Pochhammer symbols, $(x|y)_{C+1}/(x|y)_C$, is reduced to a constant. While for the ratio of Stirling numbers, such as $S^{y+1}_{x+1,\alpha}/S^y_{x,\alpha}$, can be computed quickly via caching [2].

For example, the conditional posterior for aspect a_{dn} is

$$p(a_{dn}, \mathbf{C}|\mathbf{A}^{-dn}, \mathbf{R}, \mathbf{T}, \mathbf{O}, \mathbf{C}^{-dn}, \zeta) = \frac{p(\mathbf{A}, \mathbf{R}, \mathbf{T}, \mathbf{O}, \mathbf{C}|\zeta)}{p(\mathbf{A}^{-dn}, \mathbf{R}, \mathbf{T}, \mathbf{O}, \mathbf{C}^{-dn}|\zeta)} \quad , \tag{4}$$

where the superscript $\square^{-dn}$ indicates that the target-opinion pair $\langle t_{dn}, a_{dn}\rangle$ is removed from the respective sets. It is trivial to show that the conditional posterior simplifies to ratios of Pochhammer symbols and a ratio of Stirling numbers with Equation 2 and Equation 3. The conditional posterior probability for sampling the sentiment r_{dn} can be similarly written.

Note the change in associated counts $\mathbf{C}|\mathbf{C}^{-dn}$ will be the full possible range of $+1$'s propagated up the hierarchy. So sampling $r_{dn} = r$ will increment $c_{o_{dn}}^{\phi'_{t_{dn}r}}$ and may/may-not increment $c'^{\phi'_{t_{dn}r}}_{o_{dn}}$. If it does increment $c'^{\phi'_{t_{dn}r}}_{o_{dn}}$ then it also increments $c_{o_{dn}}^{\phi_r}$, but then $c'^{\phi_r}_{o_{dn}}$ may or may-not be incremented. Sampling all these increments corresponds to sampling on a small tree of Booleans which can be done in closed form. Similarly, sampling a new $a_{dn} = a$ will increment $c_a^{\theta_d}$, and if $c'^{\theta_d}_a$ is also incremented, a new aspect cluster is created for t_{dn}.

We summarize the collapsed Gibbs sampler in Algorithm 1, and refer the interested reader to the supplementary material for detail.

7.2 Hyperparameters Sampling

During inference, we sample the hyperparameters of the PYP using an auxiliary variable sampler [38]. Moreover, we propose a novel method to update the hyperparameter b, which controls the strength of the sentiment prior. Instead of sampling the hyperparameter b (*e.g.* using the slice sampler [30]), we adopt an optimization approach since the posterior of b is highly concentrated in a small region (thin-tailed). The posterior density is given by the following equation, subject to a normalization constant.

$$p(b|\vec{c}) \propto p(b) \prod_r \prod_v \left(\frac{(1+b)^{X_{rv}}}{\sum_i \sum_j (1+b)^{X_{ij}}}\right)^{c_{rv}} ,$$

where c_{rv} is the number of times a word v is assigned a sentiment r, and $p(b)$ is the hyperprior of b. We assume a weak hyperprior for b, $b \sim \text{Gamma}(1, 1)$.

During inference, we update b to its *maximum a posteriori probability* (MAP) estimate using a gradient ascent algorithm. We opti-

Algorithm 2 Gradient Ascent Optimization for Hyperparameter b

1. Given an initial value for $b = b_0$, evaluate the gradient $l'(b_0)$.
2. Given a learning rate τ, update b to $b_i = b_{i-1} + \tau \times l'(b_{i-1})$, if the new log posterior $l(b_i)$ is lower than $l(b_{i-1})$, we halve the learning rate: $\tau := \tau/2$.
3. Repeat step 2 until b converges.

mize the log posterior $l(b) = \log(p(b|\vec{c}))$ since log is an increasing function. The gradient of the log posterior is derived as

$$l'(b) = \frac{1}{(1+b)} \sum_r \sum_v c_{rv} \left(X_{rv} - \mathbb{E}_{\phi_r}[X_r]\right) + \rho'(b) \ ,$$

where $\mathbb{E}_{\phi_r}[X_r]$ is the expected value of X_r under the probability distribution ϕ_r, and $\rho'(b)$ is the derivative of $\log p(b)$. We summarize the gradient ascent algorithm in Algorithm 2. Additionally, in the supplementary material, we present the gradient derivation and a plot of the log posteriors of b given different statistics $\vec{c}$.

8. DATA

For experiments, we perform aspect-based opinion analysis on tweets, which are characterized by their limited 140 characters text. From the *Twitter 7* dataset[5] [46], we queried for tweets that are related to electronic products such as *camera* and *mobile phones* (see the list of our query words in the supplementary material). We then remove non-English tweets with *langid.py* [21]. Moreover, since most spam tweets contain a URL, we adopt a conservative approach to remove spam by discarding tweets containing URLs. This results in a dataset of about 9 million tweets, which we name as the electronic product dataset.

Due to the lack of sentiment labels on the electronic product dataset, we make use of the Sentiment140 (Sent140) tweets[6] [8] for sentiment classification evaluation. Each Sent140 tweet contains a sentiment label (positive or negative) that are determined by emoticons. The whole corpus contains 1.6 million tweets, with half of them labeled as positive and the other half as negative.

In addition, we also use the SemEval 2013 dataset[7] [29] for evaluation. SemEval tweets are annotated on Mechanical Turk, which arguably provides better sentiment labels compared to Sent140. Since annotation is expensive, SemEval has only 6322 tweets.

8.1 Data Preprocessing

Here, we describe the preprocessing steps that we apply to tweets. Firstly, we apply Twitter NLP [31], a state-of-the-art tool for part-of-speech (POS) tagging on tweets. We then apply word normalization to clean up the tweets. We make use of the lexical normalization dictionary[8] from Han et al. [9], but modify it such that proper nouns are not normalized. For instance, words like 'iphone' and 'xbox' are not normalized, since they are the targets we are interested in. We perform normalization after POS tagging since tweets normalization degrades the performance of Twitter NLP [10].

Next, we proceed to extract target-opinion pairs from the data. Following Moghaddam and Ester [28], we apply the Stanford Dependency Parser [5] to extract dependency relations that will be used to form the target-opinion pairs. However, our approach is slightly different: we do not use the *Direct Object* (*dobj*) relation to obtain a target-opinion pair, for example, the sentence "*I like the perfect picture quality*" gives '*dobj*(like, picture quality)' and '*amod*(picture quality, perfect)', resulting in two target-opinion pairs, ⟨*picture quality, like*⟩ and ⟨*picture quality, perfect*⟩. We drop the target-opinion pair associated with *dobj* and instead use the *dobj* relation for the emotion indicator variable. Note that we use the *caseless English model* in the Stanford Dependency Parser, which works better for tweets. Additionally, since standard NLP tools perform less optimally on tweets [36], we use the POS tagging from Twitter NLP to clean up the target-opinion pairs. We note that negations like '*not*' are captured as dependency relations, the negated words are then treated as new words with the prefix '*not_*'.

[5] http://snap.stanford.edu/data/twitter7.html

[6] http://help.sentiment140.com/home

[7] http://www.cs.york.ac.uk/semeval-2013/task2/

[8] http://ww2.cs.mu.oz.au/~tim/#resources

Table 2: Emoticons and Strong Sentiment Words

Positive	Negative
:-) :o) :] :3 :c) :> =] 8) =) :} :-D ;-D :D 8-D \o/ ^_^ (^O^)/ (^^)/ :}	>:-(>:[:-(:c :@ >:(;(;-(:'-(:'(D; (T_T) (;_;) (;_:) T.T !_!
happy glad love delighted like	sad upset hate dislike angry

We determine the emotion indicator variable *via* the existence of emoticons, strong sentiment words and/or the *dobj* relation in each tweet. We simply set the emotion indicator to -1 (negative) or 1 (positive) as long as the indicators agree with one another, and unobserved otherwise. The list of emoticons used is compiled from Wikipedia[9]. We present a subset of the emoticons and strong sentiment words in Table 2, while the full list is available in the supplementary material. For Sent140 and SemEval tweets, we replace the unobserved emotion indicator by their sentiment label.

We then perform tweet aggregation, which is found to give significant improvement for LDA [24]. We group tweets that contain the same hashtag (word prefixed with # symbol) or same mention (word prefixed with @ symbol) into a single document, this allows co-occurrence within the same *tags* (our abbreviation for hashtags and mention) to be used by topic models. Grouping tweets also allows us to summarize the results for each tag, giving us a better opinion overview (see Subsection 9.3 for example). Additionally, we discard tags that occur infrequently. We note that although tweets are merged to form a larger document, the emotion indicator (variable e) is observed and stored for each individual tweet (rather than the merged document), this prevents the emotion indicator from being lost through merging.

Finally, we perform other standard preprocessing techniques to topic modeling, this consists of decapitalizing the words, removing stop words and discarding commonly occurred words and infrequent words. We define the common words as words that appear in at least 90% of the documents, and infrequent words as words that appear less than 50 times in the corpus. We randomly split the data into 90% training set and 10% test set for evaluation. We present a summary of the preprocessing pipeline in Figure 3.

8.2 Corpus Statistics

On average, we found that there are 0.69 target-opinion pair extracted per electronic product tweet. Out of the electronic tweets that contain at least one target-opinion pair, 17.9% of them contain an emotion indicator. After preprocessing, the number of unique target word tokens in the electronic product tweets is 4402, while the number of unique opinion word tokens is 25188. We present a summary of the corpus statistics for all datasets in Table 3.

[9] http://en.wikipedia.org/wiki/Kaomoji and http://en.wikipedia.org/wiki/List_of_emoticons

Table 3: Corpus Statistics

	Electronic	Sent140	SemEval
Number of tweets	∼9M	1.6M	6322
Opinion pairs per tweet	0.69	0.41	0.47
% tweets containing e	17.9	100	57.5
Target vocabulary	4402	1050	1875
Opinion vocabulary	25188	8599	813

For the electronic product tweets, the top tags are #apple, #phone, #iphone, #computer and #laptop. We note that some tags are associated with products, brands or companies, for example, #playstation and #xbox are associated with gaming products, while #sony and #canon are associated with companies. In Subsection 9.3 below, we show that aggregating hashtags allow us to have a more focused view on certain products or companies, as well as facilitating comparison between these products or companies side-by-side.

9. EXPERIMENTS AND RESULTS

In this section, we demonstrate the usefulness of TOTM for opinion mining. We evaluate TOTM quantitatively against ILDA and LDA-DP in terms of perplexity and sentiment classification. To compare the effectiveness of various sentiment lexicons, we propose a novel sentiment metric to evaluate the sentiment-opinion word distributions ϕ's. Qualitatively, we utilize TOTM for the task of opinion mining from the electronic product tweets, and show that we are able to extract various useful opinions on technological products such as iPhone.

9.1 Experiment Settings

For all the experiments, we initialize the hyperparameters of PYP to $\alpha = \beta = 0.1$ and the sentiment hyperparameter to $b = 10$, noting that the hyperparameters are optimized automatically as discussed in Subsection 7.2.

To determine the optimal number of latent aspects (A) for ILDA, we set aside 5% of the training data as development set, and select A (tested in increment of 10) such that perplexity of the development set is minimized. For a fair comparison between TOTM and ILDA, we cap the maximum number of aspects of TOTM to be that of ILDA. Our experiment finds that the number of aspects in TOTM always converges to the cap. We note that LDA-DP has only three fixed '*topics*', which is the number of sentiments.

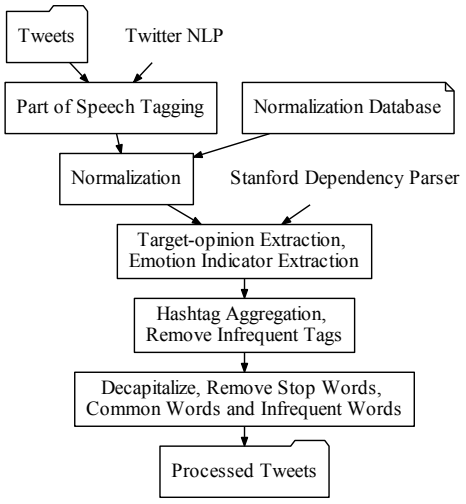


Figure 3: Preprocessing Pipeline

Table 4: Test Perplexity on Electronic Product Tweets

	Target	Opinion	Overall
LDA-DP	N/A	510.15 ± 0.08	N/A
ILDA	594.81 ± 13.61	519.84 ± 0.43	556.03 ± 6.22
TOTM	592.91 ± 13.86	**137.42** ± 0.28	**285.42** ± 3.23

Table 5: Sentiment Classification Results (%)

Sent140 Tweets	Accuracy	Precision	Recall	F1-score
LDA-DP	57.3	56.1	90.1	69.2
ILDA	54.1	56.9	55.3	55.9
TOTM	**65.0**	**61.7**	**90.2**	**73.3**
SemEval Tweets	Accuracy	Precision	Recall	F1-score
LDA-DP	52.1	65.0	58.3	61.4
ILDA	46.8	60.7	53.6	56.3
TOTM	**73.3**	**84.0**	**74.9**	**79.0**

During inference, we run the collapsed Gibbs algorithm until the convergence criteria is satisfied, defined by which the training log likelihood does not differ by more than 0.1% in ten consecutive iterations. Empirically, we find that all experiments converge within 200 iterations, indicating a good Gibbs sampling algorithm.

9.2 Quantitative Evaluations

9.2.1 Perplexity

We compute the perplexity of the test set to measure how well the models fit to the data. The perplexity is negatively related to the likelihood of the test data. Since aspect-based opinion analysis deals with two types of vocabulary, we compute the perplexity for both target words and opinion words, in this case:

$$\text{perplexity}(\mathbf{W}) = \exp\left(-\frac{\sum_{d=1}^{D}\log P(\vec{w}_d)}{\sum_{d=1}^{D} N_d}\right),$$

where $\mathbf{W}$ can be either target words $\mathbf{T}$ or opinion words $\mathbf{O}$, N_d is the number of the target-opinion pairs in document d. We also compute the overall perplexity, which is given by

$$\text{perplexity}(\mathbf{T},\mathbf{O}) = \exp\left(-\frac{\sum_{d=1}^{D}\log P(\vec{t}_d,\vec{o}_d)}{2\sum_{d=1}^{D} N_d}\right).$$

We present the perplexity result (the lower the better) for the electronic product tweets in Table 4. We present the perplexity result of Sent140 tweets and SemEval tweets in the supplementary material, for which the same conclusion can be drawn. From the perplexity results, it is clear that modeling the target-opinion pairs directly leads to significant improvement of opinion words perplexity and hence the overall perplexity. Note that LDA-DP only models the opinion words, thus we can only compare the perplexity for opinion words, we can see that its result is comparable to that of ILDA, albeit slightly better.

9.2.2 Sentiment Classification

Here, we perform a classification task to predict the polarity of the test data for Sent140 and SemEval data. We determine the polarity of a test document d by simply selecting the polarity r that gives higher likelihood in ϕ_r:

$$\text{polarity}(d) = \underset{r=\{-1,1\}}{\text{argmax}} \prod_i \phi_{r,o_{di}}.$$

Table 6: Sentiment Evaluations for the Sentiment Priors (in unit of 0.01)

	Electronic Product Tweets		*Sent140 Tweets*		*SemEval Tweets*	
	Negativity	Positivity	Negativity	Positivity	Negativity	Positivity
No lexicon	17.82 ± 1.26	17.39 ± 0.45	22.63 ± 0.96	32.31 ± 1.98	15.24 ± 1.45	21.03 ± 3.85
MPQA	**23.91** ± 0.49	31.96 ± 0.09	24.10 ± 0.49	**42.65** ± 1.02	16.88 ± 0.31	29.47 ± 0.99
SentiStrength	23.19 ± 0.08	**35.69** ± 0.33	**24.29** ± 1.07	41.26 ± 1.53	**16.94** ± 0.78	**32.17** ± 2.07

Table 7: Top Target Words for Electronic Product Tweets

Aspects (a)	Target Words (t)
Camera	camera, pictures, video camera, shots
Apple iPod	ipod, ipod touch, songs, song, music
Android phone	android, apps, app, phones, keyboard
Macbook	macbook, macbook pro, macbook air
Nintendo games	nintendo, games, game, gameboy

For simplicity, our evaluation is a binary classification task, as such, we do not include neutral tweets from SemEval data during evaluation. Note that Sent140 data does not have neutral tweets.

We present the classification *accuracy*, *precision*, *recall* and the *F1 score* in Table 5. We can see that TOTM outperforms LDA-DP and ILDA on both datasets, suggesting that our prior formulation is more appropriate than that of LDA-DP. We can also see that LDA-DP gives a better sentiment classification compared to ILDA, which does not incorporate any prior information. Note that the classification result for SemEval data is better than that of Sent140. We conjecture that this is because Sent140's sentiment labels are obtained from the emoticons, which are noisy in nature; while the sentiment labels for SemEval data is annotated.

9.2.3 *Evaluating the Sentiment Prior*

We propose a novel method to evaluate the learned sentiment-opinion phrase distributions ϕ by using another sentiment lexicon. We use the SentiWordNet lexicon for evaluation, noting that the lexicon used during training is the SentiStrength lexicon.

Unlike SentiStrength, the SentiWordNet lexicon provides two values for each word. We name them the positive affinity Z_v^+ and negative affinity Z_v^- for a given word v, they ranged from 0 to 1. For example, the word '*active*' has a positive affinity of 0.5 and a negative affinity of 0.125; while '*supreme*' has a positive affinity or 0.75 and a negative affinity of 0.

Given the affinities, we propose the following sentiment score to evaluate an opinion word distribution ϕ_r:

$$Score(\phi_r, Z) = E_{\phi_r}[Z] = \sum_{v=1}^{V_o} Z_v \phi_{rv} \ ,$$

where Z is either Z^+ or Z^-, the positive or negative affinity. The sentiment score is also the expected sentiment under the opinion word distribution.

Here, we evaluate ϕ_{-1} with negative affinity Z^- and ϕ_1 with positive affinity Z^+. We compare the sentiment scores between the cases when a sentiment lexicon is used and when it is not. Additionally, we also make use of the MPQA Subjectivity lexicon for sentiment prior (during training) and compare the sentiment evaluation against the SentiStrength lexicon. We present the result in Table 6. As we can see, it is clear that incorporating prior information results in huge improvement in the sentiment score. Also, the priors for SentiStrength are slightly better than MPQA on average. We note that optimizing the hyperparameter b is very important, as it relieves us from tuning the hyperparameter manually. To illustrate, the optimized b converges to 2.59 on the electronic product tweets, while on Sent140 and SemEval dataset, the b converges to 1.85 and 0.71 respectively. We also find that, in our tests, an incorrectly chosen b can lead to a bad result.

Table 8: Opinion Analysis of Target Words with TOTM

Target (t)	+/−	Opinions (o)
phone	−	**dead** damn stupid bad crazy
	+	**mobile smart** good great f***ing
battery life	−	terrible poor bad horrible **non-existence**
	+	good **long** great **7hr ultralong**
game	−	**addictive** stupid free **full addicting**
	+	great good awesome favorite **cat-and-mouse**
sausage	−	silly **argentinian cold** huge stupid
	+	**hot grilled** good **sweet** awesome

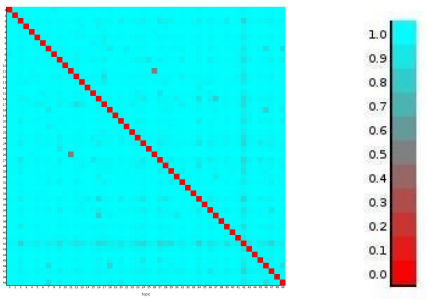
* Words in **bold** are more specific and can only describe certain targets.

9.3 Qualitative Analysis and Applications

9.3.1 *Analyzing Word Distributions*

First, we inspect the clustering of target words by TOTM and ILDA, noting that LDA-DP does not model the target words. We calculate the pair-wise Hellinger distance between each document-aspect distribution and found that the aspects are distinctive. Hellinger distance is commonly used to measure the dissimilarity between two probability distributions. The Hellinger distances between all pairs of aspect distributions from TOTM is displayed as a heat map in Figure 4, we can see that the distances between the topics are high, indicating that there is no duplicated aspect. We note that the heat map for ILDA is similar and hence not presented here. We also display an extract of the top target words from TOTM in Table 7. Our empirical examination on the aspect-target word distributions suggest that both TOTM and ILDA perform well in clustering the target words.

We then look at the opinion phrase distributions ϕ's. In ILDA and LDA-DP, the opinion words are generated conditioned on the latent sentiment labels, meaning that the opinion word is assumed to be independent to the target word given the sentiment; while in TOTM, the opinion word distributions are modeled given the senti-



(a) Heat Map (b) Legend

Figure 4: Pair-wise Hellinger Distances for Aspects (Colored)

Table 9: Aspect-based Opinion Comparison between Sony, Canon and Samsung

Brands	Sentiment	Aspects / Targets' Opinions		
		Camera	Phone	Printer
Canon	−	*camera* → expensive small bad *lens* → prime cheap broken		*printer* → obscure violent digital *scanner* → cheap
	+	*camera* → great compact amazing *pictures* → great nice creative		*printer* → good great nice *scanner* → great fine
Sony	−	*camera* → big crappy defective *lens* → vertical cheap wide	*phone* → worst crappy shittest *battery life* → low	*printer* → stupid
	+	*photos* → great lovely amazing *camera* → good great nice	*phone* → great smart beautiful *reception* → perfect	
Samsung	−	*camera* → digital free crazy *shots* → quick wide	*phone* → stupid bad fake *battery life* → solid poor terrible	*scanner* → worst
	+	*camera* → gorgeous great cool *pics* → nice great perfect	*phone* → mobile great nice *service* → good sweet friendly	

ment and the observed target word. The advantage of TOTM over ILDA and LDA-DP in modeling the opinion words is that it allows us to analyze the opinions in a finer grained view. For instance, we can display a list of positive and negative opinions associated to a certain target word; an extract of this result is presented in Table 8, in which we pick a few distinctive target words to show their opinion words distribution. As we can see from Table 8, despite some opinion words can generally be applied to most target words (*e.g.* good, bad), the highlighted words are more descriptive (*e.g.* addictive, fried, grilled) and can only be applied to certain target words. Such a result cannot be achieved by ILDA or LDA-DP.

9.3.2 *Comparing Opinions on Brands with TOTM*

We present an application of comparing opinions on entities or products using TOTM. Since entities and products are frequently quoted with tags, we can compare them directly by looking at the opinions associated with each tag. We present an extract of the opinion comparison between three brands (Canon, Sony and Samsung) in Table 9. This table shows that we can have a high level comparison of the camera product between these three brands. For the phone product, there are only comparison between Sony and Samsung, since Canon does not manufacture phones (or no tweet on such topic is found). Note that the entries under the aspect '*printer*' are lacking, we find that this is due to the low amount of opinion tweets on printers in the dataset.

9.3.3 *Extracting Contrastive Opinions on Products*

Although the above comparison is useful for providing a high level summary, it is also important to inspect the original tweets as they provide opinions in greater details. We use TOTM to extract tweets containing people's opinions on iPhone. In Table 10, we display an extract of contrasting tweets containing the target 'iphone' with positive or negative sentiment ($r = \{-1, 1\}$).

Table 10: Contrasting Opinions on iPhone

Positive	Negative
RT @user : the iPhone is so awesome!!! Emailing, texting, surfing the sametime! — Can do all tgat while talkin on the phone?...	@user awww thx! I can't send an email right now bc my iPhone is stupid with sending emails. Lol but I can tweet or dm u?
Ahhh! Tweeting on my gorgeous iPhone! I missed you! hehe am on my way home, put the kettle on will you pls :)	It would appear that the iPhone, due to construction, is weak at holding signal. Combine that with a bullshit 3G network in Denver.
Thanks @user for the link to iPhone vs Blackberry debate. I got the iPhone & it's just magic! So intuitive!	@user @user Ah, well there you go. The iPhone is dead, long live Android! ;)
Finally my fave lover @user has Twitter & will be using it all the time with her cool new iPhone :)	@user Finally eh? :D I think iphone is so ugly x.x

10. CONCLUSION

In this paper, we study the use of LDA-based models for opinion analysis on tweets queried with electronic product terms. This is motivated by the fact that Twitter is a popular platform for opinions and tweets are publicly available. Unlike reviews, tweets do not contain scores or ratings, they are more informal and usually accompanied by emoticons and strong sentiment words. Taking advantage of the informal nature of tweets, we designed a topic model named Twitter Opinion Topic Model (TOTM) for opinion analysis. TOTM is shown to greatly improve opinion prediction with the direct target-opinion modeling. In incorporating a sentiment lexicon into topic models, we proposed a new formulation for the topic model priors, which learns and updates given data. Our innovative formulation is shown to improve sentiment analysis significantly.

Our qualitative analysis demonstrates that opinion mining on tweets provide useful opinions on electronic products. Note that although we can obtain a large quantity of product opinions on tweets, the opinions are usually much noisier than reviews. For instance, opinions can be incidental (*e.g.* the author was just frustrated with the product that time), since it is easy and effortless to produce a tweet. As with the reviews, the opinions on tweets may not always be true. Some tweets are laden with sarcasm, making them difficult to interpret, while some others are spam containing no useful information.

We emphasize the importance of preprocessing steps. For instance, word normalization allows misspellings and abbreviations to be captured for target-opinion analysis; tweet aggregation improves aspect clustering and lets us compare different products or brands. For practical applications, filtering sarcastic tweets and spam is also important. In this paper, we have attempted to filter spam by removing tweets containing URLs. We acknowledge that although there is existing work on removing sarcastic tweets and spam [44, 23], we did not incorporate them due to the lack of publicly available software. As future work, we are interested in utilizing other word lexicons such as synonym and antonym lexicons into an LDA-based model for sentiment analysis.

11. ACKNOWLEDGMENTS

NICTA is funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. We also like to thank Scott Sanner, Shamin Kinathil, Rishi Dua and the anonymous reviewers for their feedback and comments.

12. REFERENCES

[1] S. Baccianella, A. Esuli, and F. Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, pages 2200–2204, 2010.

[2] W. Buntine and M. Hutter. A Bayesian review of the Poisson-Dirichlet process. *arXiv:1007.0296v2*, 2012.

[3] C. Chen, L. Du, and W. Buntine. Sampling table configurations for the hierarchical Poisson-Dirichlet Process. In *ECML*, pages 296–311, 2011.

[4] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using Twitter hashtags and smileys. In *COLING*, pages 241–249, 2010.

[5] M. De Marneffe, B. MacCartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *LREC*, pages 449–454, 2006.

[6] X. Ding, B. Liu, and P. Yu. A holistic lexicon-based approach to opinion mining. In *WSDM*. ACM, 2008.

[7] C. Fellbaum. *WordNet*. Wiley Online Library, 1999.

[8] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.

[9] B. Han, P. Cook, and T. Baldwin. Automatically constructing a normalisation dictionary for microblogs. In *EMNLP-CoNLL*, pages 421–432. ACL, 2012.

[10] B. Han, P. Cook, and T. Baldwin. Lexical normalization for social media text. *ACM TIST*, 4(1):5:1–5:27, Feb. 2013.

[11] Y. He. Incorporating sentiment prior knowledge for weakly supervised sentiment analysis. *ACM TALIP*, 11(2):4, 2012.

[12] M. Hu and B. Liu. Mining opinion features in customer reviews. In *AAAI*, volume 4, pages 755–760, 2004.

[13] J. Jagarlamudi, H. Daumé, III, and R. Udupa. Incorporating lexical priors into topic models. In *EACL*. ACM, 2012.

[14] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent Twitter sentiment classification. In *ACL*, pages 151–160, 2011.

[15] Y. Jo and A. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, pages 815–824, 2011.

[16] F. Li, C. Han, M. Huang, X. Zhu, Y.-J. Xia, S. Zhang, and H. Yu. Structure-aware review mining and summarization. In *COLING*, pages 653–661. ACL, 2010.

[17] T. Li, Y. Zhang, and V. Sindhwani. A non-negative matrix tri-factorization approach to sentiment classification with lexical prior knowledge. In *AFNLP*, pages 244–252, 2009.

[18] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *CIKM*, pages 375–384. ACM, 2009.

[19] B. Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on HLT*, 5(1):1–167, 2012.

[20] S. Liu, F. Li, F. Li, X. Cheng, and H. Shen. Adaptive co-training SVM for sentiment classification on tweets. In *CIKM*, pages 2079–2088. ACM, 2013.

[21] M. Lui and T. Baldwin. langid.py: An off-the-shelf language identification tool. In *ACL*, pages 25–30, 2012.

[22] D. Maynard, K. Bontcheva, and D. Rout. Challenges in developing opinion mining tools for social media. *@NLP can u tag #usergeneratedcontent*, 2012.

[23] M. McCord and M. Chuah. Spam detection on Twitter using traditional classifiers. In *Autonomic and Trusted Computing*, pages 175–186. Springer, 2011.

[24] R. Mehrotra, S. Sanner, W. Buntine, and L. Xie. Improving LDA topic models for microblogs via Tweet pooling and automatic labeling. In *SIGIR*, pages 889–892. ACM, 2013.

[25] Q. Mei, X. Ling, M. Wondra, et al. Topic Sentiment Mixture: Modeling facets and opinions in weblogs. In *WWW*, 2007.

[26] S. Moghaddam and M. Ester. Opinion Digger: An unsupervised opinion miner from unstructured product reviews. In *CIKM*, pages 1825–1828. ACM, 2010.

[27] S. Moghaddam and M. Ester. ILDA: Interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *SIGIR*, pages 665–674, 2011.

[28] S. Moghaddam and M. Ester. On the design of LDA models for aspect-based opinion mining. In *CIKM*. ACM, 2012.

[29] P. Nakov, Z. Kozareva, A. Ritter, S. Rosenthal, V. Stoyanov, and T. Wilson. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Workshop on Semantic Evaluation*, 2013.

[30] R. Neal. Slice sampling. *Ann. Statist.*, 31(3):705–767, 2003.

[31] O. Owoputi, B. O'Connor, C. Dyer, et al. Improved part-of-speech tagging for online conversational text with word clusters. In *NAACL-HLT*, pages 380–390, 2013.

[32] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, 2010.

[33] B. Pang and L. Lee. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[34] J. Pitman. Some developments of the Blackwell-Macqueen urn scheme. *Lecture Notes-Monograph Series*, 1996.

[35] A.-M. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28. Springer, 2007.

[36] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in Tweets: An experimental study. In *EMNLP*, pages 1524–1534, 2011.

[37] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[38] Y. W. Teh. A Bayesian interpretation of interpolated Kneser-Ney. *Tech Report A2/06, NUS*, 2006.

[39] Y. W. Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *ACL*, pages 985–992. ACL, 2006.

[40] Y. W. Teh and M. Jordan. Hierarchical Bayesian non-parametric models with applications. *Bayesian Non-parametrics: Principles and Practice*, pages 158–207, 2010.

[41] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *JASIST*, 61(12):2544–2558, 2010.

[42] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL08: HLT*, 2008.

[43] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120, 2008.

[44] O. Tsur, D. Davidov, and A. Rappoport. ICWSM-A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *ICWSM*, 2010.

[45] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT-EMNLP*, pages 347–354, 2005.

[46] J. Yang and J. Leskovec. Patterns of temporal variation in online media. In *WSDM*, pages 177–186, 2011.

[47] W. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing Twitter and traditional media using topic models. In *ECIR*, pages 338–349, 2011.

[48] W. Zhao, J. Jiang, H. Yan, and X. Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *EMNLP*, pages 56–65, 2010.