Accounting for the Correspondence in Commented Data

Renqin Cai
rc7ne@virginia.edu
University of Virginia
Department of Computer Science
Charlottesville, VA 22904, USA

Chi Wang
chiw@microsoft.com
Microsoft Research
Redmond, WA 98052, USA

Hongning Wang
hw5x@virginia.edu
University of Virginia
Department of Computer Science
Charlottesville, VA 22904, USA

ABSTRACT

One important way for people to make their voice heard is to comment on the articles they have read online, such as news reports and blogs. The user-generated comments together with the commented documents form a unique *correspondence* structure. Properly modeling the dependency in such data is thus vital for one to obtain accurate insight of people's opinions and attention.

In this work, we develop a Commented Correspondence Topic Model to model correspondence in commented text data. We focus on two levels of correspondence. First, to capture topic-level correspondence, we treat the topic assignments in commented documents as the prior to their comments' topic proportions. This captures the thematic dependency between commented documents and their comments. Second, to capture word-level correspondence, we utilize the Dirichlet compound multinomial distribution to model topics. This captures the word repetition patterns within the commented data. By integrating these two aspects, our model demonstrated encouraging performance in capturing the correspondence structure, which provides improved results in modeling user-generated content, spam comment detection, and sentence-based comment retrieval compared with state-of-the-art topic model solutions for correspondence modeling.

CCS CONCEPTS

•**Information systems** →**Document topic models;** *Web mining;* •**Computing methodologies** →*Learning in probabilistic graphical models;*

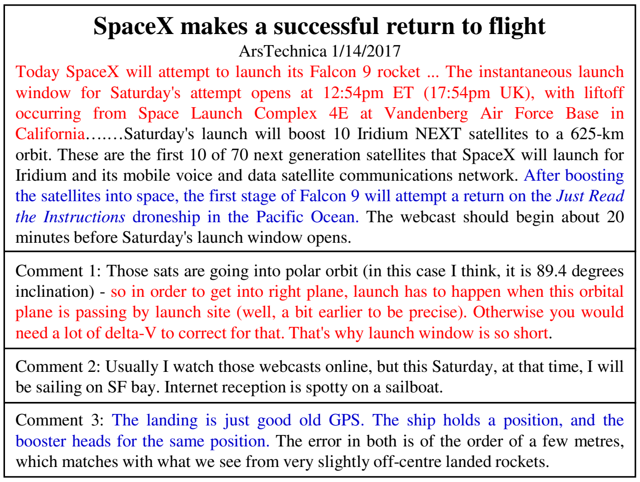
KEYWORDS

Topic models; text correspondence modeling; social media; user comments

ACM Reference format:
Renqin Cai, Chi Wang, and Hongning Wang. 2017. Accounting for the Correspondence in Commented Data. In *Proceedings of SIGIR'17, August 7-11, 2017, Shinjuku, Tokyo, Japan.*, 10 pages.
DOI: http://dx.doi.org/10.1145/3077136.3080781

1 INTRODUCTION

Modern news media websites provide commenting facilities for their readers to openly express their attention and opinions on news events. Because of readers' active participation, the user-generated commenting content shapes the general public's opinions by supplementing the news stories with contextual information and new perspectives [21, 27]. However, oftentimes, a popular news article can easily accumulate thousands of comments within a short period of time, which makes it difficult for interested users to access and digest information in such data [17, 26]. Therefore, modeling the user-generated comments with respect to the commented news articles and automatically gaining the insight of readers' opinions and attention on the news event becomes an important research topic in web data mining [7, 11, 24].



SpaceX makes a successful return to flight
ArsTechnica 1/14/2017
Today SpaceX will attempt to launch its Falcon 9 rocket ... The instantaneous launch window for Saturday's attempt opens at 12:54pm ET (17:54pm UK), with liftoff occurring from Space Launch Complex 4E at Vandenberg Air Force Base in California.......Saturday's launch will boost 10 Iridium NEXT satellites to a 625-km orbit. These are the first 10 of 70 next generation satellites that SpaceX will launch for Iridium and its mobile voice and data satellite communications network. After boosting the satellites into space, the first stage of Falcon 9 will attempt a return on the *Just Read the Instructions* droneship in the Pacific Ocean. The webcast should begin about 20 minutes before Saturday's launch window opens.

Comment 1: Those sats are going into polar orbit (in this case I think, it is 89.4 degrees inclination) - so in order to get into right plane, launch has to happen when this orbital plane is passing by launch site (well, a bit earlier to be precise). Otherwise you would need a lot of delta-V to correct for that. That's why launch window is so short.

Comment 2: Usually I watch those webcasts online, but this Saturday, at that time, I will be sailing on SF bay. Internet reception is spotty on a sailboat.

Comment 3: The landing is just good old GPS. The ship holds a position, and the booster heads for the same position. The error in both is of the order of a few metres, which matches with what we see from very slightly off-centre landed rockets.

Figure 1: An example thread of user comments on a news article about SpaceX's rocket launch.

The key to a successful understanding of such user-generated commented data is its intrinsic *correspondence* structure, i.e., readers choose the part of their most interest to comment on, and pass along some perspectives, topics or words from the news articles to their comments. Figure 1 illustrates the correspondence structure in a typical discussion thread of users' comments on a news article. The news article covers details of the rocket's launch (in red color) and return (in blue color), the satellite's orbit and mission, and webcast of the launch. Two concrete examples of correspondence are highlighted in the first and third comment in red and blue respectively. The first user comment provides detailed explanation about why the launch window is so specific and short, and the third comment explains the technique behind rocket return. We can observe that a news article may emphasize some general or objective aspects of an event, while user comments may express more specific, subjective, and supplementary information. If such

SIGIR'17, August 7-11, 2017, Shinjuku, Tokyo, Japan.
© 2017 ACM. 978-1-4503-4887-4/17/08...$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080781

correspondence structure can be automatically identified, it would help users easily retrieve comments related to a particular topic in the article, filter out irrelevant or spam comments, summarize the others' focus and opinions on the news article, and many more.

However, several challenges make the correspondence structure difficult to detect. First, there are salient stylistic and organizational distinctions between these two types of content: news articles are typically well-written and fact-oriented long stories by journalists or professionals, while comments are mostly personalized, more opinionated, and free-style short posts by average readers. The *vocabulary gap* between the news articles and user comments obscures correspondence modeling. For instance, the third comment in Figure 1 talks about the technique of rocket return, but it has almost no word overlapping with the article, as it keeps using the word "land" to explain the rocket "return" technique. Second, readers read the article before making comments, so that the background or context of their commenting content is often enclosed in the article but omitted in the comments. This further enlarges the vocabulary gap. Third, the existence of irrelevant comments further complicates correspondence modeling. For example, in Figure 1, the second comment is barely related to the main theme of the article. It is necessary to distinguish them from other relevant comments.

Researchers have attempted to resolve the first challenge using probabilistic topic modeling technique.

The basic idea of existing solutions roots in the correspondence LDA model (CorrLDA) [3], which was originally designed to model the correspondence between images and their captions. Follow-up work adopted this model as building blocks to capture the correspondence between news articles and comments [7, 17]. However, the CorrLDA-like models do not address the other two challenges discussed above. First, they assume comments' topics are a subset of the article's. This assumption generally holds in image-caption data, as a caption should be a concise summarization of the image. But in article-comment data, comments often cover topics missing in the news article, without mentioning the existence of irrelevant and spam comments. Second, existing solutions postulate that all article-comment threads share a common set of topic-word distributions. Such treatment has limited capability in capturing salient correspondence patterns which vary across threads. For example, in Figure 1, the phrase "*launch window*" from the article repeating in the first comment is a strong signal of topical correspondence, although they may not be the most typical words of that topic in the whole corpus.

In this work, we propose to address these limitations from two perspectives. At the topic level, we generate topic distributions in user comments from an article-specific Dirichlet prior. The prior consists of two parts. One part is from a common hyperparameter shared by all comments, and the other is from the topic distribution of the commented news article. This allows our model to capture the content of comments on both covered and uncovered topics of the article, while accounting for the correspondence. The impact of this prior is naturally adjusted due to the conjugacy between Dirichlet and multinomial distributions, i.e., topic assignments in shorter comments tend to get more influenced from the article, while longer comments tend to develop their own themes.

At the word level, we explicitly model the "burstiness" of word occurrences between article and comments to capture the thread-specific correspondence pattern, i.e., words from the articles repeat in their comments more often than just by chance. To the best of our knowledge, it is the first time that this phenomenon is confirmed in commented data. In particular, we choose the Dirichlet compound multinomial (DCM) distribution [9] to represent topics, capturing the tendency that the same topic manifests itself with different word distributions in different article-comment threads. Since DCM draws multinomial distributions for each article-comment thread from a corpus-wise shared Dirichlet distribution, the local word co-occurrence patterns are still connected to the global topic-word distributions.

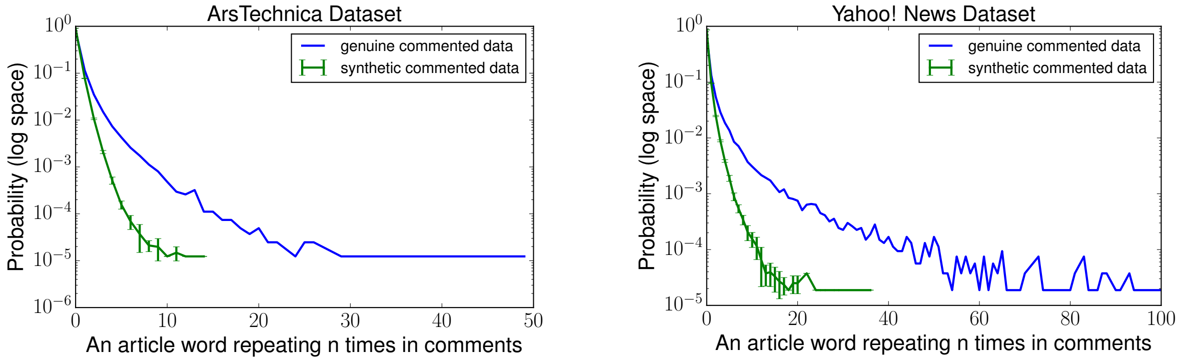
Integrating these two new components, we develop Commented Correspondence Topic Model (CCTM). Extensive experiments on two large news archives with user comments confirm the effectiveness of our model: it provides improved results in predicting unseen comments, spam comment detection, and sentence-based comment retrieval compared with state-of-the-art topic model solutions.

2 RELATED WORK

The blooming of social media enables the public to freely express their opinions on various topics [15, 23]. And one typical way is to comment on the online articles. A rich body of research has been done to discover useful knowledge buried in such user-generated commented data for facilitating various information retrieval tasks, including document retrieval, classification, summarization, and so on. Hu et al. [14] selected sentences from articles by computing their similarities to the comments for generating comment-driven news summarization. Tholpadi et al. [25] studied multi-lingual commented data to cluster comments about the same news event across different languages. Park et al. [20] utilized commented app review data to improve app retrieval performance. In general, learning accurate representations of commented data has broad potential to facilitate these information retrieval tasks.

Probabilistic topic models like LDA have achieved great success in modeling unstructured text documents [2]. Formally, a topic is modeled as a probability distribution over terms in a shared vocabulary, and a document is concisely modeled as a mixture over the topics [4, 13]. In standard topic models, the dependency among documents is loosely governed by a globally shared prior over topic distributions, e.g., Dirichlet prior. The correspondence in commented data, such as between news articles and their comments, is not considered in such models.

As an extension to LDA, CorrLDA is desinged to model the correspondence in annotated image data [3]. CorrLDA captures the correspondence structure by enforcing the topics of captions to be uniformly sampled from the topics of the corresponding images. As a result, the topic distributions in these two types of data become dependent. Follow-up research extends this modeling approach to capture correspondence among different types of text documents [10, 17]. In [7], the specific correspondence topic model (SCTM) is proposed, which introduces Dirichlet Process prior [1] into CorrLDA to fit the correspondence between news articles and user comments. However, as restricted by the inherited design from CorrLDA of uniform sampling of topics from news articles to user comments, the choices of topics in comments are strongly



(a) ArshTechinica dataset

(b) Yahoo! News dataset

Figure 2: Burstiness in commented data. Synthetic comments are sampled from a unigram language model estimated via a maximum likelihood estimator on genuine comments in each data set. The x-axis denotes the frequency of a word in an article's comments if it is observed in the article, and the y-axis denotes the corresponding the probability in log space.

constrained by the existing topics in the articles. To relax this restriction, we consider topic distributions in news articles only as a prior to topic distributions in the corresponding user comments. New topics can therefore be sampled in user comments if there is support. Similar idea has been explored in [20], but they merged all comments into a single document when modeling the correspondence, which loses fine-grained understanding in an individual comment. Furthermore, in all aforementioned solutions, the topics are globally shared among all documents, such that possible vocabulary gap and local correspondence structure cannot be recognized. In our work, we introduce Dirichlet compound multinomial (DCM) distribution [18], replacing the multinomial distribution, to model the topics and capture detailed correspondence in a thread of article and comments.

[6, 16] noticed that the first occurrence of a term in a document increases the probability of its repeated appearances in the same document. This linguistic characteristics is referred as "burstiness." To capture burstiness in a text document, DCM assumes each document is generated from a document-specific multinomial distribution of words, which is drawn from a corpus-wise Dirichlet prior. Intuitively, it postulates a "bag-of-bags-of-words" generative process for documents. DCMLDA [8] extends this modeling approach to model a document as a mixture of DCM. In this work, we further extend the concept of burstiness beyond a single document: we verified that if a word is used once in a news article, it is more likely to be used again in the corresponding user comments. This helps us model local correspondence and align it with global topic distributions. This proves to be important in spam comment detection and sentence-based comment retrieval.

3 METHODOLOGY

In this section, we first state and verify our hypothesis about burstiness of word occurrences in commented news data, and then describe the detailed design of our proposed solution, which explicitly models the topic-level and word-level correspondence. An efficient Gibbs sampler is developed to perform posterior inference, and a stochastic Expectation Maximization algorithm is utilized to estimate the model parameters.

3.1 Burstiness in Commented Data

One of our most important observations in the commented news data is that words appearing in a news article are likely to re-occur in its comments, as users quote from the news articles after reading them. It is analogous to the burstiness of word occurrences in a single document observed in prior research [6, 16]. Importantly, the burstiness of a word and its informativeness are positively correlated: more informative words are more bursty [18].

Formally, we hypothesize that the probability of a word from an article appearing multiple times in its comments is significantly larger than the probability of this word appearing in the comments with the same frequency independently. We verify this hypothesis by the following statistical test. We count the frequency of words in a given commented news corpus to estimate the probability of a word from an article appearing *n* times in its comments (i.e., maximum likelihood estimation). To estimate the probability of a word from an article *independently* appearing *n* times in comments, we generate synthetic comments. We first estimate a unigram language model based on all comments in the corpus, and generate synthetic comments for each article by sampling from the language model. In particular, we enforce the synthetic comments to have the same length as the genuine comments. As the words are drawn from a unigram language model, they are independent from each other in the synthetic comments. Then we use the same maximum likelihood estimator to estimate the probability of an article word appearing *n* times in its synthetic comments. Note that we only count words *already observed* in the articles, and therefore if burstiness exists, it is about the repetition pattern of words from articles in its corresponding comments.

We tested our hypothesis on two large real-world commented corpora, Yahoo! News and ArsTechnica Science blog data set. The detailed descriptions about these two datasets and our preprocessing procedures can be found in the experiment section. We repeated the synthetic data generation procedure for ten times and reported the mean and standard deviation of the estimated probabilities in Figure 2. As we can clearly observe from the results, the probability of a word occurring multiple times in the genuine commented data is significantly larger than that in the synthetic data. Moreover, the maximum frequency of a word repeated in the genuine commented data is much larger than that in synthetic data. These observations indicate the repeated occurrences of words in the commented data are *not* independent, and thus our hypothesis about burstiness of words holds in such data. We should note that we only generated synthetic comments while keeping the genuine news articles intact in both cases for probability estimation.

In addition, as the comments are generally short, the burstiness of article words within each single comment is negligible. To the best of our knowledge, this is the first time that the word burstiness phenomenon has been verified in the commented text data.

3.2 Commented Correspondence Topic Model

In this section, we describe the details about our Commented Correspondence Topic Model (CCTM). As motivated in Section 1, our model integrates two main ideas: topic-level and word-level correspondence modeling.

At the topic level, both articles and comments are modeled as mixtures over K topics, where the mixing proportion θ^a in article a and θ^c in comment c are drawn from their corresponding Dirichlet priors. Accordingly, the topic assignments of words in articles and comments are drawn from those mixing proportions, i.e., $z^a \sim Mul(\theta^a)$ and $z^c \sim Mul(\theta^c)$. The prior distribution for θ^a is shared across all articles in the corpus, i.e., $\theta^a \sim Dir(\alpha^A)$; but CCTM differs from previous models in how θ^c is generated. As we discussed before, the model needs to account for both relevant and irrelevant comments to the article, and allow for both covered and uncovered topics in the article to appear in the comment. For this reason,

we postulate an article-specific Dirichlet prior for θ^c, which takes the topic assignments from the corresponding article as the parameter,

$$\theta^c \sim Dir(\bar{z}^a + \alpha^C) \tag{1}$$

where $\bar{z}^a = \frac{1}{N}\sum_{n=1}^{N} z_n^a$, z_n^a is the topic assignment for n-th word in article a, and α^C is a hyperparameter vector shared by all comments in the corpus. We choose $\bar{z}^a$ rather than θ^a as the parameter for topic prior of comments, because $\bar{z}^a$ represents the actual topics observed in the article, which is presumably more accurate to capture the article's theme. Due to this prior design, the posterior topic distribution θ^c in each comment of an article is not only determined by the shared context specified in $\bar{z}^a + \alpha^C$, but also by the observed words in comments. Consequently, comments on the same article tend to have similar topic distributions, but can also have different topic choices if there is support.

At the word level, each word w in an article or a comment is generated from a multinomial distribution parameterized by ϕ^z after its topic assignment z is determined. CCTM also differs from previous models in how ϕ^z is generated. To model both the thread-specific burstiness of words and the global word co-occurrence patterns, we leverage Dirichlet compound multinomial (DCM) distribution [18] when generating ϕ^z, and enforce an article and its comments to share the same set of thread-specific topics.

Specifically, for each article-comment thread, the DCM distribution draws a set of multinomial distributions $\left\{p(w|\phi_j)\right\}_{j=1}^{K}$, which are drawn from a set of Dirichlet priors parametrized by $\{\beta_j\}_{j=1}^{K}$:

$$\phi_j \sim Dir(\beta_j), j = 1, 2, ..., K.$$

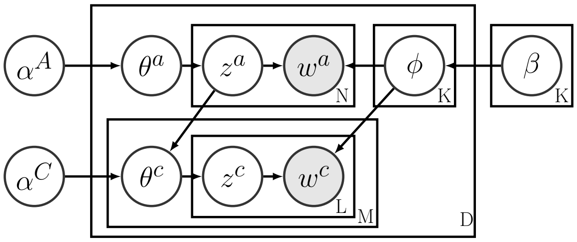


Figure 3: Graphical model representation of CCTM. Dark and light circles represent observable and latent random variables, and plates denote repetitions. Arrows encode dependency relation among the random variables.

The Dirichlet priors are shared in the corpus. Under this modeling assumption, an article-comment thread can have its own word distribution for each topic, and these local word distributions are loosely related across the article-comment threads by the commonly shared Dirichlet priors.

To be explicit, given the topic assignment of a word, the conditional probability of this word in a corpus is computed as,

$$\begin{aligned} p(w|z=k,\beta) &= \int p(w|\phi_k)p(\phi_k|\beta_k)d\phi_k \\ &= \prod_{d}^{D} \frac{\Gamma(A_{wkd}^{WKD} + \beta_{wk})}{\Gamma(\beta_{wk})} \frac{\Gamma(\sum_w \beta_{wk})}{\Gamma(\sum_w A_{wkd}^{WKD} + \beta_{wk})} \end{aligned} \tag{2}$$

where A_{wkd}^{WKD} denotes the number of times word w in the article-comment thread d is assigned to topic k. From Eq. (2), we can observe the probability of a word drawing a particular topic in an article-comment thread is *conditionally independent* from other article-comment threads given Dirichlet parameters $\{\beta_j\}_{j=1}^{K}$ in our model. Therefore, the repetition of words between an article and its comments becomes a very strong signal of content correspondence, even if this word is not globally representative of the topic.

Integrating these two model specifications, the generative process of the commented data in CCTM can be described as follows:

For each article a_d, $1 \le d \le D$,
1. For each topic k, $1 \le k \le K$, sample the word distribution $\phi_k^d \sim Dir(\beta_k)$;
2. Sample topic proportion $\theta^{a_d} \sim Dir(\alpha^A)$;
3. For each word $w_n^{a_d}$, $1 \le n \le N$ in a_d,
I. Sample topic $z_n^{a_d} \sim Multi(\theta^{a_d})$;
II. Sample word $w_n^{a_d} \sim Multi(\phi_{z_n^{a_d}}^d)$;
For each comment c_{dm}, $1 \le m \le M_{a_d}$ of a_d,
1. Sample topic proportion $\theta^{c_{dm}} \sim Dir(\alpha^C + \bar{z}^{a_d})$;
2. For each word $w_l^{c_{dm}}$, $1 \le l \le L_{c_{dm}}$ in c_{dm},
I. Sample topic $z_l^{c_{dm}} \sim Multi(\theta^{c_{dm}})$;
II. draw word $w_l^{c_{dm}} \sim Multi(\phi_{z_l^{c_{dm}}}^d)$.

where D is the number of articles, N is the number of words in the article a_d, M_{a_d} is the number of comments associating with the article a_d, and $L_{c_{dm}}$ is the number of words in the comment c_{dm}. Using the language of graphical models, this generative process can be depicted in Figure 3.

3.3 Inference & Parameter Estimation

In CCTM, the latent variables of interest are the topic assignments $\{z_n^a\}_{n=1}^N$ for each article a, $\{z_l^c\}_{l=1}^L$ for each comment c, and $\{\phi_k\}_{k=1}^K$ for each article-comment thread. The discrete topic assignments represent articles and comments in a concise topic space and reflect the correspondence between these two types of documents. And the topic-word distributions $\{\phi_k\}_{k=1}^K$ reveal the instantiation of topics in each article-comment thread. However, exact inference of these random variables in CCTM is computationally intractable. Thanks to the conjugacy between Dirichlet and multinomial distributions, we develop an efficient collapsed Gibbs sampler to perform posterior inference of these random variables in each article and comment.

In traditional topic models, the corpus-level hyperparameters are usually set manually [12]. In CCTM, the hyperparameters, i.e., $\{\alpha^A, \alpha^C, \beta_{k=1:K}\}$ convey important semantics. Since DCM is used for modeling topics, $\{\beta_k\}_{k=1}^K$ represent K global topics. α^A and α^C reflect the topical focus in news articles and comments respectively. Manually tuning them is difficult, and we propose to estimate them from data. Our estimation is based on the maximization of complete data likelihood $p(\boldsymbol{w}, z|\alpha^A, \alpha^C, \beta_{k=1:K})$ [5]. Specifically, we adopt the Monte Carlo Expectation Maximization algorithm. In E-step, we fix the hyperparameters, and perform posterior inference of the topic assignments in articles and comments. In M-step, we fix the topic assignments and maximize the complete data likelihood. To collect independent samples from the sampling chain, we only keep samples every five iterations, i.e., thinning the sampling chain. Besides, samples from the beginning of the sampling chain (i.e., the burn-in period) may not accurately represent the desired distribution. We only keep the posterior samples after the burn-in period (in our experiment, we discarded the first 20% of samples).

In the following, we provide the detailed inference and parameter estimation procedures in E-step and M-step.

• **E-step**. Due to the Dirichlet-multinomial conjugacy, the latent variables of θ^a, θ^c and $\{\phi_k\}_{k=1}^K$ can be marginalized out in the resulting posterior distribution in each article-comment thread. This leaves us an efficient collapsed Gibbs sampler to infer the topic assignments $\{z_n^a\}_{n=1}^N$ in article a and $\{z_l^c\}_{l=1}^L$ in comment c.

In the following derivation, we define a set of sufficient statistics to simplify our description of conditional probabilities for sampling. A_{wkd}^{WKD} represents the count of word w assigned to topic k in the article a_d and all its comments $\left\{c_{dm}\right\}_{m=1}^{M_{a_d}}$. A_{kd}^{KD} denotes the number of words assigned to topic k in the article a_d, and C_{kmd}^{KMD} denotes the number of words assigned to topic k in article a_d's comment c_{dm}. The special symbol '$-n$' denotes the word w_n is excluded when computing the corresponding sufficient statistics for sampling.

$$
\begin{aligned}
p(z_n^{a_d} = j|w_n^{a_d}, z_{-n}^{a_d}, w_{-n}^{a_d}, \boldsymbol{z}^{c_d}, \boldsymbol{w}^{c_d}, \alpha^A, \alpha^C, \boldsymbol{\beta}) \\
\propto \frac{\beta_{w_n^{a_d} j} + A_{wjd,-n}^{WKD}}{\sum_{w'}(\beta_{w'j} + A_{w'jd,-n}^{WKD})}(\alpha_j^A + A_{jd,-n}^{KD}) \\
\times \prod_{m=1}^{M_{a_d}} \prod_{k=1}^{K} \frac{\Gamma\left(\alpha_k^C + \frac{A_{kd,-n}^{KD} + 1(k=j)}{\sum_k A_{kd}^{KD}} + C_{kmd}^{KMD}\right)}{\Gamma\left(\alpha_k^C + \frac{A_{kd,-n}^{KD} + 1(k=j)}{\sum_k A_{kd}^{KD}}\right)}
\end{aligned}
\tag{3}
$$

Based on these notations, the conditional probability of assigning topic j to word $w_n^{a_d}$ in article a_d is computed as Eq.(3), where $\boldsymbol{w}^{c_d}$ and $\boldsymbol{z}^{c_d}$ represent the words and corresponding topic assignments in all comments associated with article a_d. And the conditional probability of assigning topic j to word $w_l^{c_{dm}}$ in comment c_{dm} is,

$$
\begin{aligned}
& p(z_l^{c_{dm}} = j|w_l^{c_{dm}}, z_{-l}^c, w_{-l}^c, z^a, w^a, \alpha^A, \alpha^C, \beta) \\
& \propto \frac{\beta_{w_l^{c_{dm}} j} + A_{wjd}^{WKD}}{\sum_{w'}(\beta_{w'j} + A_{w'jd,-l}^{WKD})} \frac{\alpha_j^C + \frac{A_{jd}^{KD}}{\sum_k A_{kd}^{KD}} + C_{jmd,-l}^{KMD}}{1 + \sum_k(\alpha_k^C + C_{kmd,-l}^{KMD})}
\end{aligned}
\tag{4}
$$

The topic-level correspondence between an article and its comments is clearly depicted in the above sampling equations. As stated in Eq. (3), the topic assignments in an article are determined by not only the words in this article, but also words in all its comments (specified by the Gamma ratio function). Especially, this Gamma ratio function encourages the topic that is frequently mentioned in the comments (as we have C_{kmd}^{KMD} on the numerator) to be observed in the article. And this relation is more clearly encoded in the second part of Eq. (4): the topic assignments in the article serve as pseudo count in each comment's topic proportion, which promotes major topics of the article in their comments.

The word-level correspondence is also reflected in the sampling equations. As denoted in the first part of Eq. (3) and (4), the conditional probability of word w sampled from topic j depends on the word-topic allocation in the current article a_d, its comments $\{c_{dm}\}_{m=1}^{M_{a_d}}$, and the global Dirichlet prior. This promotes the local co-occurrence patterns of words, which might not be necessarily frequent in the whole corpus, and encourages assigning the same topic to the repeated words, i.e., capturing burstiness.

Once the sampling chain converges in each article-comment thread, we can easily estimate other latent variables as follows,

$$\theta_k^{a_d} \propto \alpha_k^A + A_{kd}^{KD} \tag{5}$$

$$\theta_k^{c_{dm}} \propto \alpha_k^C + \frac{A_{kd}^{KD}}{\sum_k A_{kd}^{KD}} + C_{kmd}^{KMD} \tag{6}$$

$$\phi_{wk}^d \propto \beta_{wk} + A_{wkd}^{WKD} \tag{7}$$

We should note that as $\{\phi_k\}_{k=1}^K$ are article-specific, the above sampling procedures can be readily parallelized with respect to the articles. This enables CCTM to easily scale to large collections of commented data.

• **M-step.** Based on the posterior inference results in each article-comment thread, the maximum likelihood estimation of the hyperparameters $\{\alpha^A, \alpha^C, \beta_{k=1:K}\}$ can be independently performed over the complete data log-likelihood function. We only illustrate the estimation procedure for $\{\beta_k\}_{k=1}^K$, as the same approach directly applies to α^A and α^C.

Specifically, we estimate $\{\beta_k\}_{k=1}^K$ by maximizing the complete log-likelihood function of CCTM,

$$
\begin{aligned}
L(\beta) = & \sum_{d,w,k} \left(\log\Gamma(A_{wkd}^{WKD} + \beta_{wk}) - \log\Gamma(\beta_{wk})\right) \\
& + \sum_{d,k} \left(\log\Gamma(\sum_w \beta_{wk}) - \log\Gamma(\sum_w A_{wkd}^{WKD} + \beta_{wk})\right)
\end{aligned}
$$

This complete log-likelihood function indicates that the updates of $\{\beta_k\}_{k=1}^K$ are independent across K topics, and thus we can update these parameters in parallel. As no closed-form solution exists for the above optimization problem, we appeal to the fixed point iteration method to iteratively optimize it [19]. The update equation of β_{wk} is:

$$\beta_{wk}^{new} = \beta_{wk}^{old} \frac{\sum_d \Big(\psi(A_{wkd}^{WKD} + \beta_{wk}^{old}) - \psi(\beta_{wk}^{old})\Big)}{\sum_d \Big(\psi(\sum_w A_{wkd}^{wKD} + \sum_w \beta_{wk}^{old}) - \psi(\sum_w \beta_{wk}^{old})\Big)}$$

where β_{wk}^{old} is the value obtained in last M-step and $\psi(\cdot)$ is the first order derivative of the log Gamma function. The update of $\{\beta_k\}_{k=1}^K$ helps recondition global topics from the inferred local topics. Importantly, such an iterative solution is guaranteed to converge to a stationary point of the complete log likelihood function [19]; and for the Dirichlet distribution, the global maximum is the only stationary point.

4 EXPERIMENTAL EVALUATIONS

Our evaluation is conducted in multiple ways. First, we present case study in Section 4.1 to investigate the quality of inferred topics from CCTM. Second, we use perplexity to compare the capability of our model with several topic model based solutions in predicting unseen commented data in Section 4.2. Third, we study two important applications based on the inferred correspondence structure, namely spam comment detection and targeted comment retrieval in Section 4.3 and 4.4 respectively.

We used ArsTechnica Science [1] technical blog dataset and the Yahoo! News dataset provided in [7] as our evaluation corpus. In the ArsTechnica dataset, comments are manually annotated with their corresponding sentences in the articles. The annotations serve as ground-truth for us to evaluate the learned correspondence by different solutions. The Yahoo! News dataset was originally unannotated. We used crowd sourcing to annotate a set of pooled correspondence mappings between articles and comments based on different algorithms' inference results. Standard text preprocessing steps were performed on these two datasets, including stopword removal, stemming and normalization. We removed articles and comments whose length is shorter than 5 words after the pre-processing. The basic statistics of the two evaluation datasets after these pre-processing steps are shown in Table 1.

Table 1: Evaluation Corpus Statistics.

DataSet	ArsTechnica	Yahoo! News
#Articles	501	651
#Comments	2897	32285
#Words	216769	476497
Vocalbulary Size	6053	8755

In our evaluation, we include the following three topic model based solutions for correspondence modeling as our baselines.

- Latent Dirichlet Allocation (LDA) [4]. Each article and comment are modeled as independent documents.
- Correspondence LDA (CorrLDA) [3]. It is an extension of LDA model, where topics in comments are uniformly drawn from topics in the corresponding article.
- Specific Correspondence Topic Model (SCTM) [7]. It is a state-of-the-art model on capturing correspondence in commented data. In this model, each article is split into sentences, and the topics of words in comments are uniformly drawn from topic assignments in the selected sentences.

Besides these three existing models, we also include two variations of CCTM to evaluate the impact of the two main ingredients of CCTM: topic-level prior setting for flexible topic modeling, and word-level DCM for thread-specific topic-word distributions.

- CCTM-. It removes the DCM component from CCTM, i.e., using one shared set of multinomial distributions to represent topics for the entire corpus like the standard topic models.
- CorrLDA+. It adds the DCM component to CorrLDA, but the topics in comments are still uniformly sampled from topic assignments in the corresponding article.

4.1 Case Study

We first investigate the quality of topic-word distributions learned in CCTM, and then study the inferred topical representation of articles and comments. As similar results were obtained in Yahoo! News dataset, we only discuss the results obtained in ArsTechnica dataset in this section.

CCTM learns two levels of topics. The Dirichlet prior parameters $\{\beta_k\}_{k=1}^K$ in CCTM represent global topics, and $\{\phi_k\}_{k=1}^K$ in each article-comment thread represent thread-specific instantiation of global topics. Figure 4 visualizes three example global topics and corresponding local topics from two different article-comment threads, using a tree structure representation. The root nodes of the trees represent the words selected by $\{\beta_k\}_{k=1}^K$ from three global topics, and the leaf nodes of each tree represent the corresponding topics from two randomly selected article-comment threads, where the words are selected by the thread-specific topic-word distribution $\{\phi_k\}_{k=1}^K$. Article title corresponding to each leaf node is shown under the node. In each topic node, top 10 words are shown.

Take "topic 3" in Figure 4 as an example. The top words from β_3 indicate that it is a health-related topic, while the top words from the two threads' corresponding topic ϕ_3^1 and ϕ_3^2 are also related to health but with different emphases. The first thread concentrates on "*antibiotics*" and "*bacteria*", and the second focuses on "*HIV infection*" and "*HIV treatment*". And these topical words are all strongly related to the title of these two articles and the corresponding discussion content in the articles and comments. Due to space limit, we cannot display full articles' and comments' content, but we used word clouds to highlight the keywords in the article and two selected comments for two article-comment threads. Similar observations can also be found in other randomly selected topics and threads in the figure. This result suggests that the local and global topics extracted by CCTM are meaningful, especially the local topics capture fine grained variations of topic-word distributions. The learned relation between global topics and thread-specific topics is meaningful in the commented data.

We further visualize the content of two selected article-comment threads (thread 2 and 6) using word clouds and illustrate the corresponding topic distributions in Figure 4. The content of article 2 is mostly about "virus, hiv, infection", which match top words in ϕ_3^2. Consistently, the inferred topic distribution of article 2 shows that topic 3 occupies the main body of that article. Likewise, the topic distribution of comment 2_2 suggests that this comment talks

[1] http://arstechnica.com/

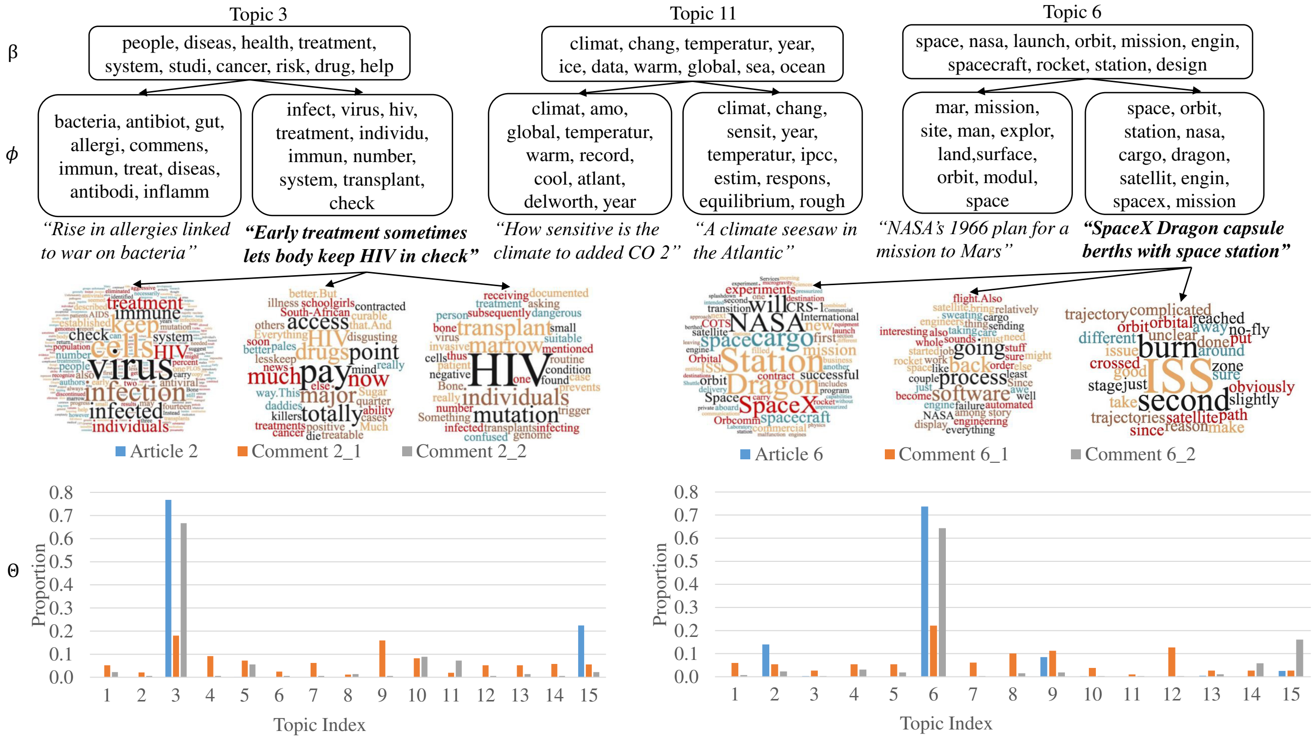


Figure 4: Illustration of the topics learned by CCTM in ArsTechnica dataset. In each box of the first level, top 10 words are selected from a global topic encoded by β_k; and in the second level, top 10 words from six randomly selected article-comment threads by ϕ_k accordingly. The article's title is labeled under each leaf node. Word clouds are used to to highlight the content of selected articles and comments on the second level. The inferred topic distributions in articles and comments are shown at the bottom of this figure.

about the same topic, which can be confirmed from its word content "HIV, mutation, transplant" highligted in the word cloud. On the other hand, comment 2_1 has fewer words related to "health, HIV", but its content spreads out on words like "pay," "access," and "point". Consistently, the inferred topic distribution of comment 2_1 is less concentrated on topic 3. This comment actually covers topics that are rarely mentioned in article 2, such as topic 4 and 9. These observations confirm that CCTM is able to identify diverse topic compositions in comments by not restricting the topics in comments as a subset of those in articles, but only treating topics in articles as a prior to the topics in comments.

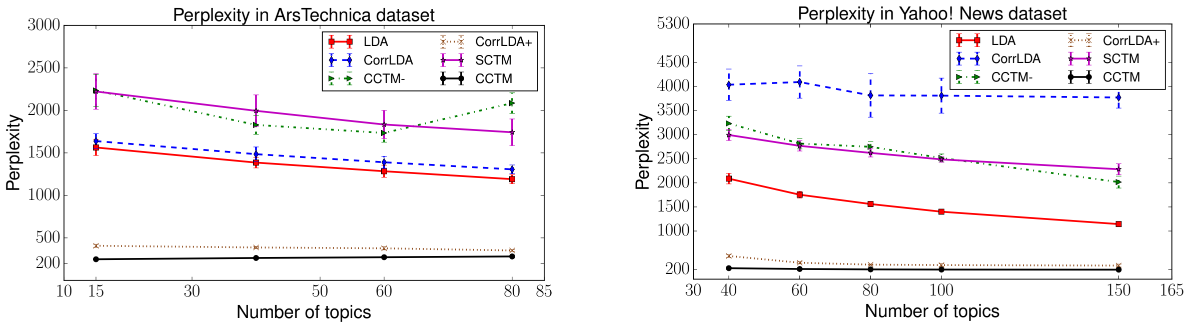
4.2 Perplexity

Perplexity is a standard metric evaluating the predictive power of a probabilistic model, and it is computed as the exponentiation of the model's entropy on held-out data. A low perplexity indicates the model is good at predicting unseen data.

To compute perplexity in the commented data, we preserve a portion of article-comment threads as training data for estimating the model parameters, such as $\{\alpha^A, \alpha^C, \beta_{k=1:K}\}$ in CCTM. Then we apply the learned model on the rest part of collection for topic inference and likelihood computation. We varied the number of topics, performed ten-fold cross validation in each train/test separation, and reported the resulting perplexity on both evaluation datasets in Figure 5. One important parameter in all topic model based solutions is the number of topics. In our experiment, we varied the number of topics in all models and used perplexity as the metric to select the optimal settings. We found in most of cases, this setting did not affect the relative comparison of these models' perplexity results too much in both of our evaluation datasets. Consequently, we set the topic size to 15 in ArsTechnica dataset and set it to 80 in Yahoo! News dataset for all models in all our reported experiments.

From the results, we can observe that CCTM outperformed all baselines. The major reason of this low perplexity is that CCTM utilizes the DCM distribution to model topics, which provides sufficient flexibility for the article-comment thread specific topics to exploit local word co-occurrence patterns. But all the other topic models (except CorrLDA+) have to use globally shared topics in modeling the articles and their comments, which limits their ability to model unseen data. This conclusion can be further confirmed by CorrLDA+, where DCM distribution replaces the global multinomial distribution to model topics. Its perplexity is significantly lower than other baselines.

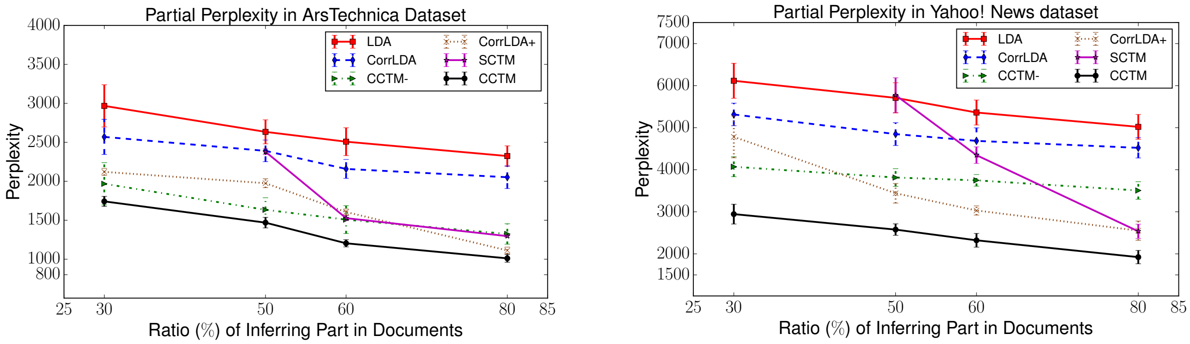
However, this standard definition of perplexity does not consider the whole story about correspondence modeling in the commented data: in such held-out testing data, the whole article-comment thread is available for topic inference. A good correspondence modeling method should be able to better predict the unseen document, immediately when some portion of the document becomes available. To make a more comprehensive comparison, we utilize partial perplexity [22, 29] to measure the performance of these models.



(a) ArshTechinica dataset

(b) Yahoo! News dataset

Figure 5: Perplexity on ArshTechinica and Yahoo! News datasets.



(a) ArshTechinica dataset

(b) Yahoo! News dataset

Figure 6: Partial Perplexity on ArshTechinica and Yahoo! News datasets.

To calculate partial perplexity, we further divide each article and comment in the test set into two parts. One part is used to infer the topic proportion of the document, a.k.a. *inferring part*, and the other part is used to compute the perplexity, a.k.a. *perplexity part*. This metric can avoid overfitting the test data, because the perplexity part is not used for inference.

We vary the ratio of inferring part to have a thorough insight about the performance of these models. At each ratio setting, perplexity is obtained via a ten-fold cross validation in each model. The results are shown in Figure 6.

Again CCTM achieved the best predictive capability against all baselines. It implies that our model characterizes the commented data more accurately due to better modeling of the correspondence structure. In addition, there is a clear gap between CCTM and its two simplified variants: CCTM- and CorrLDA+. That suggests both topic-level prior design and word-level DCM design account for the improvement over existing approaches. We can also notice that the impact of these two designs varies with respect to the ratio of inferring part. For example, on Yahoo! News dataset, when 30% of document content (including both article and comment) is used to infer the topic distribution, the gap between CorrLDA and CorrLDA+ is smaller than the gap between CorrLDA+ and CCTM, and the perplexity of CorrLDA+ is worse than CCTM-. That means the topic-level prior design plays a more important role than the local topic-word distribution design when only a few words are observed. This is expected that with fewer observations, the local word repetition pattern cannot be fully observed, and the topic assignment in comments gets more influence from the article, as the articles tend to be longer than comments in general. On the other hand, when 80% of document content is used to infer the topic distribution, the gap between CorrLDA and CorrLDA+ becomes larger than that between CorrLDA+ and CCTM, and the perplexity of CorrLDA+ is better than CCTM-. It means that when sufficient amount of words are observed, the thread-specific word burstiness pattern becomes more prominent, and thus the influence of DCM on perplexity is enhanced. The same conclusion can be drawn from both ArsTechnica and Yahoo! News datasets.

4.3 Detecting spam comments

Accurately filtering spam comments is an important task in assisting users digest the commented data. In ArsTechnica dataset, comments are exhaustively annotated with the corresponding sentences in the articles. As a result, we treat the comments without any aligned sentence in the article as irrelevant to the discussion thread, or spam comments. We assume spam comments should be more distinct from the article than the normal ones in terms of topical relatedness.

We adopt cosine similarity to measure the distance between the topic distribution in a comment and that in an article. The smaller the cosine similarity is, the more likely a comment is a spam. To study the performance of spam detection under different similarity thresholds, we report the precision and recall curve of each model in Figure 7, where ten-fold cross validation is performed.

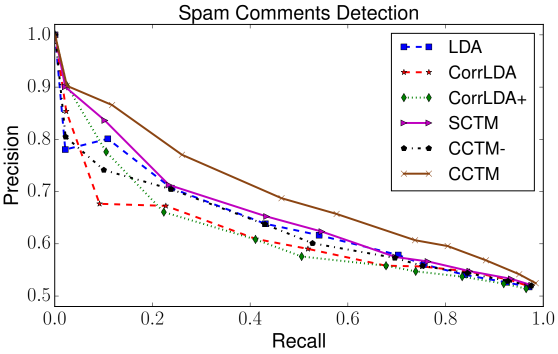


Figure 7: Performance of spam comment detection.

In this dataset, 1483 of 2897 comments are not aligned with any sentence in the article, and therefore are considered as spams. We can observe that at every recall level, CCTM achieved improved precision against all other models. CorrLDA restricts the topics of comments only to the existing topics in articles; but with the existence of spam comments, this restriction undermines the quality of learned topic distribution in comments and makes it difficult to recognize the spam comments. LDA independently models articles and comments, so that it can hardly capture the correspondence in commented data due to the vocabulary gap. Both CCTM- and CorrLDA+ performed worse than CCTM. CCTM- does not utilize the local word burstiness pattern, so that it cannot recognize the comments that use globally less popular words to describe the same topic in the articles. As a result, its precision tends to be lower at the same recall level. CorrLDA+ fails to assign unseen topics in an article to its comments although CorrLDA+ is able to capture the local topics with globally less popular words, and thus fails to detect spams as well.

SCTM is the best alternative model, which has a background topic to accommodate irrelevant comments [7]. However, as topics are globally shared in SCTM, it cannot leverage the thread-specific word-level correspondence and thus performed worse than CCTM. In addition, we can notice that SCTM has competitive precision when the recall is low, but the precision drops very quickly with increasing recall. Its precision gap to our CCTM model increases quickly too. When higher recall is required, the difficulty in detecting spam comments increases. A slower decrease in precision of our model means it outperforms other models not only on detecting obvious spam comments, but also on subtle ones via learning more accurate topical representation of comments and articles.

4.4 Retrieving comments for targeted sentences in articles

Retrieving relevant comments for a given sentence in a news article helps readers to track others' opinions and acquire complementary information about the article content. We consider a sentence from an article as a query and rank all the associated comments by the likelihood of generating this sentence given the inferred topics in comments [28]. We believe better correspondence modeling leads to better retrieval performance. In order to verify this, we employed three standard ranking metrics, Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and Precision at 3 (P@3), to measure the retrieval performance of different models.

Table 2: Retrieval performance in ArsTechnica dataset.

Model	MAP	NDCG	P@3
LDA	0.605	0.708	0.295
CorrLDA	0.592	0.698	0.288
CCTM-	0.600	0.704	0.290
CorrLDA+	0.598	0.703	0.290
SCTM	0.616	0.716	0.296
CCTM	0.632*	0.728*	0.312*

* p-value<0.05

Table 3: Retrieval performance in Yahoo! News dataset.

Model	MAP	NDCG	P@3
LDA	0.639	0.782	0.520
CorrLDA	0.643	0.799	0.541
CCTM-	0.662	0.793	0.555
CorrLDA+	0.613	0.764	0.486
SCTM	0.587	0.753	0.486
CCTM	0.678*	0.816*	0.590*

* p-value<0.05

Specifically, the likelihood of generating a sentence s from a comment c can be computed as $p(s|c) = \prod_{w \in s} \sum_z p(w|z)p(z|c)$, where $p(z|c)$ represents the posterior topic distribution in the comment and $p(w|z)$ represents the word distribution of topic z. For LDA, CorrLDA, CCTM- and SCTM, $p(w|z)$ is obtained from word distributions in global topics. And for CorrLDA+ and CCTM, $p(w|z)$ is obtained from the word distributions in local topics.

In ArsTechnica dataset, we used existing annotations as ground-truth for relevance judgment. After removing sentences which are not annotated with any relevant comments, we have 2193 annotated sentences as queries. The results are reported in Table 2 and paired t-test is performed between the best and second best performing algorithms under each performance metric.

For the Yahoo! News dataset, there was no annotation. We used Amazon Mechanical Turk to obtain the relevance judgment of comments to sentences in a set of randomly selected news articles (as this data set is too large to perform exhaustive annotation). In particular, we chose the annotation candidates by pooling different algorithms' inferred correspondence pairs between a sentence and a comment. Each algorithm picks top 5 comments for 56 selected sentences from 19 news articles. To make the selected sentences representative, we kept the first sentence of the selected article selected as it usually serves as an abstract of the news report. And we also included the sentence that was associated with the largest pooled comment set. This indicated the models disagreed on the mapping for this sentence; and therefore it best differentiated different algorithms' ranking quality. Crowd sourcing workers were asked to give relevance label of each sentence-comment pair in five levels including (1) bad - totally irrelevant; (2) fair - not quite relevant; (3) good - somehow relevant; (4) excellent - relevant; and (5) perfect - exact match. Then we treat "bad" and "fair" as irrelevant while regarding the rest as relevant in relevance based ranking metrics (i.e., MAP and P@3). For each comment-sentence pair, we required at least three out of five workers to agree on the judged relevance, otherwise we would ignore the annotated pair. Majority vote was used to determine whether a comment is relevant to the sentence. As a result, we obtained 786 valid annotations for 428 comments with respect to 48 sentences. In average each sentence was ranked against 9 comments. The ranking performance of models is reported in Table 3, and the same t-test was performed to confirm the statistical significance of the comparison.

From the results, we can observe that CCTM achieves the best ranking performance under these three metrics on both datasets. SCTM performs the 2nd best in ArcTechnica dataset, and CCTM- is the runner-up in Yahoo! News dataset. Both of them underperformed CCTM due to their inability to leverage the local word burstiness patterns. In detail, we can find that CorrLDA was penalized in this retrieval evaluation, because of its overly restricted correspondence assumption between articles and comments. Since LDA model does not model the correspondence structure, it ranked comments solely based on the globally shared topic distribution and thus performed worse than CCTM.

5 CONCLUSION

In this paper, we developed a Commented Correspondence Topic Model to model correspondence structure in commented data. Both topic-level and word-level correspondence are explicitly captured by introducing the topic assignments in commented articles as the prior to their comments' topic proportions and utilizing Dirichlet compound multinomial distribution to capture the local word repetition patterns. And, for the first time, we performed hypothesis test to verify the phenomena of word burstiness also exists in the commented data. Empirical evaluations on two large text corpora confirmed the effectiveness of the proposed model in correspondence modeling, and its utility in mining such user-generated comments. Although we took commented news data as case study, the developed model can be potentially applied to many other types of text data with correspondence structure, such as forum discussions, social media comments and product reviews.

Currently, our model does not directly model the relationship among comments, which are not independent in practice as users might form groups and discuss via commenting on each other's posts. It is necessary to capture the correspondence at this level as well. In addition, our solution does not model sentences in articles, which limits its resolution in recognizing fine-grained correspondence between articles and comments. In our future work, we will add sentence structure into correspondence modeling. Furthermore, our empirical evaluation in this work mostly focused on commented news data; but the developed solution is not limited to such data. It is necessary to explore its applicability in other types of user-generated commented data, and identify new insights of correspondence structure there.

6 ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments. This paper is based upon work supported by the National Science Foundation under grant IIS-1553568.

REFERENCES

[1] Charles E Antoniak. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The annals of statistics* (1974), 1152–1174.

[2] David M Blei. 2012. Probabilistic topic models. *Commun. ACM* 55, 4 (2012), 77–84.

[3] David M Blei and Michael I Jordan. 2003. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 127–134.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[5] Gilles Celeux, Didier Chauveau, and Jean Diebolt. 1996. Stochastic versions of the EM algorithm: an experimental study in the mixture case. *Journal of Statistical Computation and Simulation* 55, 4 (1996), 287–314.

[6] Kenneth W Church and William A Gale. 1995. Poisson mixtures. *Natural Language Engineering* 1, 02 (1995), 163–190.

[7] Mrinal Kanti Das, Trapit Bansal, and Chiranjib Bhattacharyya. 2014. Going beyond Corr-LDA for detecting specific comments on news & blogs. In *Proceedings of the 7th ACM international conference on Web search and data mining*. ACM, 483–492.

[8] Gabriel Doyle and Charles Elkan. 2009. Accounting for burstiness in topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 281–288.

[9] Charles Elkan. 2006. Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution. In *Proceedings of the 23rd international conference on Machine learning*. ACM, 289–296.

[10] Kosuke Fukumasu, Koji Eguchi, and Eric P Xing. 2012. Symmetric correspondence topic models for multilingual text analysis. In *Advances in Neural Information Processing Systems*. 1286–1294.

[11] Giorgos Giannopoulos, Ingmar Weber, Alejandro Jaimes, and Timos Sellis. 2012. Diversifying user comments on news articles. In *International Conference on Web Information Systems Engineering*. Springer, 100–113.

[12] Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences* 101, suppl 1 (2004), 5228–5235.

[13] Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 50–57.

[14] Meishan Hu, Aixin Sun, and Ee-Peng Lim. 2008. Comments-oriented document summarization: understanding documents with readers' feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 291–298.

[15] Andreas M Kaplan and Michael Haenlein. 2010. Users of the world, unite! The challenges and opportunities of Social Media. *Business horizons* 53, 1 (2010), 59–68.

[16] Slava M Katz. 1996. Distribution of content words and phrases in text and language modelling. *Natural Language Engineering* 2, 01 (1996), 15–59.

[17] Zongyang Ma, Aixin Sun, Quan Yuan, and Gao Cong. 2012. Topic-driven reader comments summarization. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. ACM, 265–274.

[18] Rasmus E Madsen, David Kauchak, and Charles Elkan. 2005. Modeling word burstiness using the Dirichlet distribution. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 545–552.

[19] Thomas Minka. 2000. Estimating a Dirichlet distribution. (2000).

[20] Dae Hoon Park, Mengwen Liu, ChengXiang Zhai, and Haohong Wang. 2015. Leveraging user reviews to improve accuracy for mobile app retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 533–542.

[21] Kristen Purcell, Lee Rainie, Amy Mitchell, Tom Rosenstiel, and Kenny Olmstead. 2010. Understanding the participatory news consumer: How Internet and cell phone users have turned news into a social experience. Pew Internet & American Life Project. (2010).

[22] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. AUAI Press, 487–494.

[23] Clay Shirky. 2011. The political power of social media: Technology, the public sphere, and political change. *Foreign affairs* (2011), 28–41.

[24] Alexandru Tatar, Jérémie Leguay, Panayotis Antoniadis, Arnaud Limbourg, Marcelo Dias de Amorim, and Serge Fdida. 2011. Predicting the popularity of online articles based on user comments. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*. ACM, 67.

[25] Goutham Tholpadi, Mrinal Kanti Das, Trapit Bansal, and Chiranjib Bhattacharyya. 2015. Relating Romanized Comments to News Articles by Inferring Multi-Glyphic Topical Correspondence.. In *AAAI*. 311–317.

[26] Manos Tsagkias, Wouter Weerkamp, and Maarten De Rijke. 2009. Predicting the volume of comments on online news stories. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 1765–1768.

[27] Manos Tsagkias, Wouter Weerkamp, and Maarten De Rijke. 2010. News comments: Exploring, modeling, and online prediction. In *European Conference on Information Retrieval*. Springer, 191–203.

[28] Xing Wei and W Bruce Croft. 2006. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 178–185.

[29] Aonan Zhang, Jun Zhu, and Bo Zhang. 2013. Sparse online topic models. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 1489–1500.