Subtle Topic Models and Discovering Subtly Manifested Software Concerns Automatically

Mrinal Kanti Das† MRINAL@CSA.IISC.ERNET.IN
Suparna Bhattacharya BSUPARNA@IN.IBM.COM
IBM Research-India

Chiranjib Bhattacharyya† CHIRU@CSA.IISC.ERNET.IN
K. Gopinath† GOPI@CSA.IISC.ERNET.IN

†Department of Computer Science and Automation,
Indian Institute of Science, Bangalore, India

Abstract

In a recent pioneering approach LDA was used to discover *cross cutting concerns*(CCC) automatically from software codebases. LDA though successful in detecting *prominent* concerns, fails to detect many useful CCCs including ones that may be heavily executed but elude discovery because they do not have a strong prevalence in source-code. We pose this problem as that of discovering topics that rarely occur in individual documents, which we will refer to as *subtle topics*. Recently an interesting approach, namely focused topic models(FTM) was proposed in (Williamson et al., 2010) for detecting rare topics. FTM, though successful in detecting topics which occur *prominently* in very few documents, is unable to detect subtle topics. Discovering subtle topics thus remains an important open problem. To address this issue we propose *subtle topic models*(STM). STM uses a generalized stick breaking process(GSBP) as a prior for defining multiple distributions over topics. This hierarchical structure on topics allows STM to discover rare topics beyond the capabilities of FTM. The associated inference is non-standard and is solved by exploiting the relationship between GSBP and generalized Dirichlet distribution. Empirical results show that STM is able to discover subtle CCC in two benchmark code-bases, a feat which is beyond the scope of existing topic models, thus demonstrating the potential of the model in automated concern discovery, a known difficult problem in Software Engineering. Furthermore it is observed that even in general text corpora STM outperforms the state of art in discovering subtle topics.

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

1. Introduction

Hierarchical Dirichlet process (HDP) (Teh et al., 2007) is one of the most widely used topic models. Recall that, HDP places a Dirichlet process (DP) prior over potentially infinite number of topics at corpus level. Subsequently, it uses a DP prior over the topics for each document, and each document level DP is distributed as the corpus level DP. Though HDP is extremely successful in discovering topics in general, it fails to discover topics which occur in very few documents, often referred as rare topics. This inability stems from the fact that HDP inherently assumes that a frequent topic will on average occur frequently within each document, leading to a positive correlation between proportion of a topic in an article and prevalence of a topic in the entire corpus.

This important problem has partially been addressed in (Williamson et al., 2010). By using Indian Buffet process (IBP), (Williamson et al., 2010) defined a compound DP namely ICD to decorrelate document wise prevalence and corpus wide proportion. ICD was applied on focused topic models (FTM) to detect rare topics which are prominently placed in very few documents.

Consider the corpus, proceedings of NIPS, 2005[1]. Because of limited number of papers on supervised classification, an HDP based approach fails to identify topics related to supervised classification but FTM detects this easily(see supplementary material for more details).

However, there are some topics which are not only rare across the documents but also rarely appear within a document. Under these situations FTM will fail to discover them. A case in point is, a topic related to neuromorphic engineering about *cochlear modelling*. The topic has been discussed only in one paper(Wen & Boahen, 2005). In addition to that, the main theme (cochlear) is rarely explicitly mentioned (5% of sentences) in that paper. Therefore, the topic is assigned an extremely low probability making it extremely difficult for even FTM to detect. This phenomena is not specific only to scientific corpora, but is also observed in other text corpora too. We studied the speeches of *Barack Obama* from July 27, 2004 till October 30, 2012[2], a span of eight years. We observe that, in this corpus there are two speeches on *Carnegie Mellon University*(CMU). Those speeches were given when he visited CMU on 2 June, 2010 and 24 June 2011. It is a difficult task for FTM to detect this topic as "Carnegie Mellon" is contained not only in two documents(which is rare), but also present in less than 10% of sentences in those two documents. These examples show that, discovering topics which rarely occur in individual documents still remain an unsolved problem. To this end, we propose to study the discovery of *subtle*[3] *topics*, which rarely occur in the corpus as well as in individual documents.

An immediate motivation for studying subtle topics is the automatic discovery of cross cutting concerns in software codes. Latent Dirichlet Allocation(LDA) (Blei et al., 2003) has been applied in software analysis to automatically discover topics that represent software concerns(Baldi et al., 2008). The use of topic models for this problem is attractive because unlike most other state-of-the-art concern identification techniques it is neither limited by apriori assumptions about the nature of concerns to look for nor by the need for human input and other sources of information beyond the source code. However, in framework based softwares, important program concerns can have such a subtle presence in the code that existing topic models fail to detect them.

[1]nips.cc/Conferences/2005

[2]www.americanrhetoric.com/barackobamaspeeches.htm

[3]The dictionary meaning of *subtle* is *difficult to detect*, which motivates the name.

In Berkeley-DB, a widely used software code base, the cross-cutting concern involved in the updation of various book-keeping counts is not easy to detect as the counters are named as `nWaits`, `nRequests`, `nINsCleanedThisRun` ... etc which do not contain the word "count". Such concerns that are expressed subtly in the source code cannot be ignored as they may be sources of high resource usage or support a critical program functionality. For example, *Verify* is a useful cross-cutting concern in Berkeley-DB, yet it is too subtle in the code for any of the existing models, including FTM, to recognize it.

Contributions: In this paper we propose the *subtle topic models*(STM) which has the ability to detect topics those occur very rarely in individual documents. HDP and FTM use a single distribution over topics for a document and we have observed that makes it difficult for them to detect subtle topics. In order to give importance to subtle topics within a document, we propose to split the co-occurrence domain inside a document by using multiple distributions over topics for each document. It is non trivial to select a proper prior over these topic vectors. We use generalized stick breaking process (GSBP) (Ishwaran & James, 2001) to address this issue. Using GSBP, STM allows the topic vectors to be shared across the document and the proportions over the topic vectors to be independent of each other which is essential in modeling subtle topics as explained in detail later. The inference problem due to GSBP is not standard. We propose to solve this by utilizing the relationship between GSBP and generalized Dirichlet distribution ($\mathcal{GD}$) and subsequently conjugacy between $\mathcal{GD}$ and the multinomial distribution. We believe that this process and the associated inference procedure is novel and is of independent interest to the Bayesian non-parametric community.

The most significant contribution in terms of the utility of STM lies in its ability to detect subtly manifested concerns in software programs, a known hard problem in software engineering. The results obtained here thus mark a breakthrough in this area. In addition, STM finds subtle topics from proceedings of NIPS, 2005 and speeches of Barack Obama since 2004, that shows its ability to do well on general text corpora.

Structure of the paper: The paper is organized as follows. Section 2 discusses the application of topic models in discovering concerns in software code bases. In section 3, we present the proposed model, while section 4 describes the inference procedure. Experimental study has been covered in section 5.

2. Topic models for detecting Software concerns

Software concerns are features, design idioms or other conceptual considerations that impact the implementation of a program. A concern can be characterized in terms of its intent and extent(Marin et al., 2007). A concern's intent is defined as its conceptual objective (or topic). A concern's extent is its concrete representation in software code, i.e. the source code modules and statements where the concern is implemented. Program concerns may be *modular*, i.e. implemented by a single source file or module, or *cross-cutting*, i.e. dispersed across several code modules and interspersed with other concerns.

Identifying and locating concerns in existing programs is an important and heavily researched problem in software (re)engineering (Robillard, 2008; Savage et al., 2010a; Marin et al., 2007; Eaddy et al., 2008; Revelle et al., 2010). Yet, it remains hard to automate completely in a satisfactory fashion. Typical concern location and aspect mining techniques are semi-automatic: some use manual query patterns and some generate seeds automatically based on structural information (Marin et al., 2007). Both approaches have the restriction that they tend to be driven by some prior expectation or search clues about the concern(s) of interest either in terms of the concerns' intent (e.g seed word patterns, test cases), or about the concerns' extent (e.g. fan-in analysis).

Recently it was shown that LDA can automatically detect *prominent* cross-cutting concerns (Baldi et al., 2008; Savage et al., 2010b) quite successfully without these restrictions.

Although the LDA approach works well for surfacing concerns (including CCCs) that have a statistically significant manifestation in the source code (a large extent), it can miss interesting CCCs (e.g. concerns that are executed heavily and thus impact runtime resource usage) just because they may not have a prominent presence in source code. This is especially likely in framework based code where all underlying module sources may not be available, and a concern's extent may include a small percentage of statements in source code files to be analyzed. Or even when a concern's extent is not all that small, it can elude detection because of the subtle presence of representative words that reflect its intent.

Consider the example of *Verify*, an important CCC in Berkeley-DB. According to a published manual analysis that includes a fine grained mapping of Berkeley-DB code concerns (available at (Apel et al., 2009; Kastner et al., 2007)), this CCC occurs individually as a main concern and also has 7 derivative concerns (combination of multiple concerns). See supplementary material for more details.

However, this concern is surprisingly hard to detect not just by LDA/HDP but even by FTM and MG-LDA(Titov & McDonald, 2008). The concern's extent is not particularly small but its statements are spread across files and contain the internals of operations performed to verify different structures. Thus the word "verify" occurs in only a small fraction of these statements. It is very challenging to surface such subtle traces of the concern's intent automatically without relying on any apriori information. Despite FTM's strength in detecting rare topics, FTM fails at this task as well because even in the file with the strongest presence of the *verify* concern, the word is reflected in less than 10% of the statements in that file. Thus, detecting subtly manifested concerns remain to be a challenging open task.

3. Subtle Topic Models

In this section we present subtle topic models (STM), designed to detect subtle topics which are rarely present across the corpus as well as within documents. We will briefly discuss generalized stick breaking process before describing STM.

3.1. Generalized stick breaking process (GSBP)

Under the generalized stick breaking process framework, any $\mathcal{P}$ is a stick breaking random measure if it is of the following form.

$$\mathcal{P} = \sum_{j=1}^{J} \rho_j \delta_{\beta_j}(.) \;\; \rho_1 = v_1, \;\; \rho_j = v_j \prod_{l<j}(1 - v_l) \qquad (1)$$

where $v_j \sim Beta(\tau_j, \iota_j)$[4], and β_js are independently chosen from a distribution H. δ_{β_j} denotes a discrete measure concentrated at β_j. By construction, $0 \leq \rho_j \leq 1$, and $\sum_{j=1}^{J} \rho_j = 1$ almost surely, where J can be finite or infinite. When $\tau_j = 1$, $\forall j$ and $\iota_j = \iota$, $\forall j$, and $J \to \infty$, then it reduces to $DP(\iota H)$. The two parameter Poisson-DP (Pitman-Yor process) corresponds to the case when $J \to \infty$, $\tau_j = 1 - \tau$, and $\iota_j = \iota + j\tau$ with $0 \leq \tau < 1$ and $\iota > -\tau$. For more discussion see (Ishwaran & James, 2001).

Here we are interested in the situation when $J < \infty$, for which to ensure $\sum_{j=1}^{J} \rho_j = 1$ one needs to set

[4] $\sim$ denotes "distributed as"

$v_J = 1$. We will utilize one interesting property of this finite dimensional stick-breaking process – random weights ρ_js defined in this manner are also generalized Dirichlet ($\mathcal{GD}$) distributed.

3.2. Subtle topic models

We consider a dataset as $\{\{\{w_{din}\}_{n=1}^{N_{di}}\}_{i=1}^{S_d}\}_{d=1}^{D}\}$, where D is the number of documents in the corpus, S_d is the number of sentences in document d, N_{di} being the number of words in sentence i of document d. In addition, let us denote the number of words in a document by N_d.

In STM, for each document d, we propose to have $J_d \geq 1$ number of distributions over topics. Topics denoted by β_ks are shared across the corpus. We assume a distribution over these J_d topic vectors at sentence level, denoted by ρ_{di} for sentence i in document d. Note that, a distribution over the topic vectors at document level will lead to the problem of having high probability for those topic vectors which are popular in the document.

3.2.1. Selecting prior over topic vectors

There are various options in choosing J_d, ρ_{di} and a prior distribution over ρ_{di}. The simplest possibility is that: $J_d = J$, $\forall d$, and $\rho_{di} \sim Dirichlet(\tau)$, τ being a J dimensional vector. The problem with a fixed J is that it can not model the fact that, the documents with higher S_d (or N_d) in general have higher probability of being more incoherent than those with smaller S_d. In order to avoid this issue, one can use $J_d \sim Poisson(S_d)$. However, this will make expected value of J_d large for documents with large S_d. That in turn increases the chance of documents with large S_d to be incoherent in most of the cases which is undesirable. However, due to the rich getting richer property HDP is not suitable in this case. On the other hand, using ICD in this case will make it difficult to learn as the content of a sentence is too small.

Noting that, J_d can be at most the number of sentences S_d in document d(when ρ_{di} has 1 in one component and zero else where and each ρ_{di} is different for different i), we set $J_d = S_d$. Then, we use GSBP as described in section 3.1 to construct ρ_{dij}s as follows. For document d, $i = 1, \ldots, S_d$ and $j = 1, \ldots, S_d - 1$

$$v_{dij} \sim Beta(\tau_j, \iota_j) \qquad (2)$$
$$\rho_{di1} = v_{di1}, \;\; \rho_{dij} = v_{dij} \prod_{l<j} (1 - v_{dil})$$

with $v_{diS_d} = 1$. Let us denote the above process as $GSBP_{S_d}(\tau, \iota)$, where τ and ι are $S_d - 1$ dimensional vectors of parameters. Note that, $\sum_{j=1}^{S_d} \rho_{dij} = 1$ as $1 - \sum_{j=1}^{S_d-1} \rho_{dij} = \prod_{l=1}^{S_d-1}(1 - v_{dil})$. Due to this construction, S_d is the upper limit and not the exact number of distributions over topics per document. Therefore, although there is a possibility of higher value of J for a larger document but selecting higher indexed topic vectors are discouraged.

As discussed earlier, with proper parameter setting, finite GSBP can be treated as truncated-DP or truncated-PYP (Pitman-Yor process). DP or PYP can also be used alternatively which does not affect rest of the model. However, GSBP is a more flexible distribution and is better suited for small sentences encountered in software datasets.

3.2.2. Construction of topic vectors

We denote the distributions over topics in document d as $\{\theta_{dj}\}$ for $j = 1, 2, \ldots, S_d$. HDP is not a suitable prior for θ_{dj} as we need the distribution over topics to be uncorrelated to the document level topic proportions and with each other as much as possible. Therefore, ICD seems to be a more appropriate choice here. We use two parameter IBP(Griffiths et al., 2007) to sample binary random vectors γ_{dj} and then we sample θ_{dj}s from ICD as follows. For, $j = 1, \ldots, S_d$, and $k = 1, \ldots, K$

$$\gamma_{djk} \sim Bernoulli(\pi_k), \theta_{djk} \sim Dirichlet(\alpha 1_K . \gamma_{dj})$$

where $\pi_k \sim Beta(\frac{\mu\delta}{K}, \delta)$. 1_K denotes a K-dimensional vector with all one and "." denotes component wise (Hadamard) product. δ is a repulsion parameter, with same expected number of topics the variability among γ_{dj}s across j increases when δ increases, and when $\delta = 1$ it reduces to standard IBP. K is the truncation level and (Doshi-Velez et al., 2009) shows that the probability of γ_{djk} to be 1 for any j is very low if K is sufficiently high.

Using the above two constructions we get the base distribution corresponding to a sentence, as follows

$$G_{di} = \int d\rho_{di} \sum_{j=1}^{S_d} \int d\theta_{dj} \sum_k \rho_{dij}\theta_{djk}\; p(\rho_{di})p(\theta_{dj})\; \delta_{\beta_k}$$

This forms a dependent Dirichlet process where the β_ks are shared across all the sentences in all the documents and the θ_{dj}s are shared across all the sentences within a document. STM assumes the generative process as described in Algorithm 1.

Algorithm 1 Generative process of STM

for $k = 1, 2, \ldots$ **do**
  draw topic $\beta_k \sim Dirichlet(\eta 1_V)$
  topic selection prob $\pi_k \sim Beta(\frac{\mu\delta}{K}, \delta)$
end for
for documents $d = 1$ **to** D **do**
  sample number of sentences $S_d \sim Poisson(\varrho)$
  for distribution over topics $j = 1, \ldots, S_d$ **do**
    sample $\gamma_{djk} \sim Bernoulli(\pi_k)$, $k = 1, \ldots$
    sample $\theta_{dj} \sim Dirichlet(\alpha 1_K . \gamma_{dj})$
  end for
  for sentences $i = 1, \ldots, S_d$ **do**
    sample $\rho_{di} \sim GSBP_{S_d}(\tau, \iota)$
    for words $n = 1, \ldots, N_{di}$ **do**
      select $b_{din} \sim mult(\rho_{di})$
      sample topic $z_{din} \sim mult(\theta_{db_{din}})$
      sample word $w_{din} \sim mult(\beta_{z_{din}})$
    end for
  end for
end for

4. Posterior Inference

We use the Gibbs sampling approach to sample the latent variables using the posterior conditional distribution. The main challenges in the inference procedure are due to the binary random vectors γs, and the generalized stick breaking process(GSBP) variables vs. We sample the binary random vectors considering truncated-IBP. A discussion on the effect of truncation can be found in (Doshi-Velez et al., 2009). Inference procedure due to GSBP is relatively unexplored area and not straightforward. We utilize the fact that GSBP is equivalent of *generalized Dirichlet distribution*($\mathcal{GD}$)(Wong, 1998). The benefit we draw from this relationship is that, like Dirichlet distribution $\mathcal{GD}$ is also conjugate to the multinomial distribution. This approach makes the inference very simple as we describe next.

We collapse the conditional distributions by integrating out the topic distributions (β), distributions over topics (θ), Bernoulli parameters (π_k) and distribution over topic vectors for each sentences (ρ_{da}). We however, sample topic assignment variables zs, and bs along with binary vector γs.

We will use notation for counts as follows. d is the document index, a is the sentence index and i is the word position index. n represents the counting variable and indices are put in the subscript, where "." represents marginalization. Super-script denotes that in all counts the current word is excluded (we do not repeat this in the text follows). Thus, $n^{-dai}_{\ldots k w_{dai}}$ is the number of times word type w_{dai} is associated with topic k. $n^{-dai}_{\ldots k.}$ represents the number of times topic k is used in the whole corpus. $n^{-dai}_{d.b_{dai}k.}$ is the number of times topic k and b_{dai} are used. $n^{-dai}_{d.b_{dai}..}$ denotes the number of times b_{dai} is used. $n^{-dai}_{daj..}$ is number of times topic vector indexed by j is used, and $n^{-dai}_{da\ldots}$ is the number of words in the sentence. K is the truncation level for topics. For the sake of brevity, in the following text we do not put all the variables in the conditional hoping that is easy to track following the generative process (Algorithm 1).

Sampling z and γ: The conditional probability of topic assignment of word i at sentence a in document d can be expressed as:

$$\begin{aligned} & p(z_{dai} = k | \mathbf{w}, \mathbf{z}^{-dai}) \\ \propto\ & p(w_{dai} | z_{dai} = k, \mathbf{z}^{-dai}) p(z_{dai} = k | \mathbf{z}^{-dai}) \\ =\ & \frac{\eta + n^{-dai}_{\ldots k w_{dai}}}{V\eta + n^{-dai}_{\ldots k.}} \ \frac{\gamma_{db_{dai}k}\alpha + n^{-dai}_{d.b_{dai}k.}}{\sum_k \gamma_{db_{dai}k}\alpha + n^{-dai}_{d.b_{dai}..}} \end{aligned} \tag{3}$$

Notice that, we need to infer only γ, and b in order to assign topics. Note that, γ contain binary selection values. Therefore, if $n^{-dai}_{d.jk.} > 0$, then a.s. posterior probability of γ_{djk} to be one is 1. Otherwise:

$$\begin{aligned} & p(\gamma_{djk} = 1 | \mathbf{z}, \gamma^{-djk}) \\ \propto\ & p(z_d | \gamma_{djk} = 1, \gamma^{-djk}) p(\gamma_{djk} = 1 | \gamma^{-djk}) \\ \propto\ & \frac{\Gamma(\sum_{s \neq k} \gamma_{djs}\alpha + \alpha)}{\Gamma(\sum_{s \neq k} \gamma_{djs}\alpha + \alpha + n^{-dai}_{d.j..})} \frac{\sum_r \sum_l \gamma_{rlk} + \frac{\mu\delta}{K}}{\sum_r \sum_l 1 + \frac{\mu\delta}{K} + \delta} \end{aligned} \tag{4}$$

Sampling b: From the relation between $GSBP$ and $\mathcal{GD}$ we get that, if ρ_{di}s are constructed as Eq. 2, then they are equivalently distributed as $\mathcal{GD}$ and the density of ρ_{di} is:

$$f_{\rho_{di}} = \prod_{j=1}^{S_d - 1} \frac{\rho_{dij}^{\tau_j - 1} (1 - \sum_{l=1}^{j} \rho_{dil})^{\kappa_j}}{B(\tau_j, \iota_j)} \tag{5}$$

where $B(\tau_j, \iota_j) = \frac{\Gamma(\tau_j)\Gamma(\iota_j)}{\Gamma(\tau_j + \iota_j)}$. $\kappa_j = \iota_j - \iota_{j+1} - \tau_{j+1}$ for $j = 1, 2, \ldots, S_d - 2$ and $\kappa_{S_d - 1} = \iota_{S_d - 1} - 1$. Note that, $\rho_{diS_d} = 1 - \sum_{l=1}^{S_d - 1} \rho_{dil}$. Note that, by setting $\iota_{j-1} = \tau_j + \iota_j$, $2 \leq j < S_d$, $\mathcal{GD}$ reduces to standard Dirichlet distribution.

Now using the conjugacy between $\mathcal{GD}$ and multinomial we integrate out ρs and vs. If $\rho_{da} \sim \mathcal{GD}_{S_d - 1}(\tau_1, \ldots, \tau_{S_d - 1}, \iota_1, \ldots, \iota_{S_d - 1})$, and b_{daj}s are

sampled from $mult(\rho_{da})$, then the posterior distribution of ρ_{da} given b_{daj}s is again a $\mathcal{GD}$ with density $\mathcal{GD}_{S_d-1}(\bar{\tau}_1, \ldots, \bar{\tau}_{S_d-1}, \bar{\iota}_1, \ldots, \bar{\iota}_{S_d-1})$, where $\bar{\tau}_j = \tau_j + n_{daj}^{-dai}, \bar{\iota}_j = \iota_j + \sum_{l=j+1}^{S_d} n_{dal}^{-dai}$. Thus we compute conditional $p(b_{dai} = j | b^{-dai}, \tau, \iota)$, for $j < S_d$ as

$$\frac{\tau_j + n_{daj}^{-dai}}{\tau_j + \iota_j + \sum_{r=j}^{S_d} n_{dar}^{-dai}} \prod_{l<j} \frac{\iota_l + \sum_{s=l+1}^{S_d} n_{das}^{-dai}}{\tau_l + \iota_l + \sum_{s=l}^{S_d} n_{das}^{-dai}}$$

and $p(b_{dai} = S_d | b^{-dai}, \tau, \iota) = 1 - \sum_{l=1}^{S_d-1} p(b_{dai} = l | b^{-dai}, \tau, \iota)$. Notice that, the stick breaking property of $GSBP$ is clearly visible here. The posterior probability of selecting a topic vector for a word can be found to be as below:

$$\begin{aligned} & p(b_{dai} = j | \mathbf{b}^{-dai}, \mathbf{z}, \gamma_{dj}) \qquad (6) \\ \propto \; & p(z_{dai} | b_{dai} = j, \mathbf{z}^{-dai}, \gamma_{dj}) p(b_{dai} = j | \mathbf{b}^{-dai}) \\ = \; & \frac{\gamma_{djz_{dai}} \alpha + n_{d.jz_{dai}.}^{-dai}}{\sum_k \gamma_{djk} \alpha + n_{d.j..}^{-dai}} \; p(b_{dai} = j | \mathbf{b}^{-dai}) \end{aligned}$$

Equations 3, 4, 6 together form the inference procedure of STM.

Discussion: Note that, when $J = 1$, we get a model equivalent to FTM, and that way we get an alternative inference procedure for FTM. Recall that, in (Williamson et al., 2010) the binary vectors are integrated out using approximation for computing conditional for topic assignment variables zs, and the binary vectors are sampled to compute π_ks. We have observed that, both these alternatives are equally well, however sampling the binary vectors makes the inference simpler with the cost of marginally slower convergence rate (matches up in likelihood in about 100 iterations) in case of truncated IBP.

5. Empirical Study

In this section we empirically study the proposed model STM on a special task of finding out subtle cross-cutting concerns from software repositories. In addition we apply STM on two text datasets which are apparently rich of subtle topics. This section is organized as follows. First we explain challenges related to empirical evaluation and our approach under the limited scope. Then we describe the baselines followed by the datasets used in the evaluation. Next, we discuss our results in two subsections followed by a short discussion on the empirical findings[5].

[5]For relevant resources see mllab.csa.iisc.ernet.in/stm.

5.1. Evaluation approach

We evaluate STM on two aspects, (1) modeling abilities by using perplexity and topic coherence. (2) ability to discover subtle topics. For the first case, we will use standard metrics. However, it is not easy to evaluate on the second aspect. Unlike semantic coherence, it is difficult for a human to judge subtlety of a topic by looking at few top words. Moreover, subtlety is relative to the dataset i.e. a topic may be subtle with respect to a dataset, but that may be prominent in some other dataset. As an alternative to human-judgment, we can check how good a model can find out some known or pre-defined subtle topics. But, it is difficult to find a dataset with a set of pre-defined subtle topics as gold standard and then to compare against that. In the given condition of unavailability, we manually create our gold-standard as explained later and provide the complete list in the supplementary.

5.2. Baselines

For evaluation, we compare with HDP, FTM[6] and MG-LDA(Titov & McDonald, 2008). MG-LDA has the ability to discover local topics which might be missed by HDP or FTM. Although, subtle topics may not localize properly inside a document, it is useful to benchmark STM against MG-LDA.

5.3. Dataset & pre-processing

NIPS-05: This is a corpus of 207 accepted papers from the proceedings of Neural Information Processing Systems, 2005 [7] (Globerson et al., 2007).

Obama-speech: Collection of public speeches by Barack Obama since July 27, 2004 till October 30, 2012[8] that comprises 142 articles which are transcribed directly from audio.

BerkeleyDB: We have selected Berkeley DB Java Edition as our software dataset(Apel et al., 2009). As of 2012, Berkeley DB is the most widely used database toolkit in the world[9], and it is known to have a wide range of cross-cutting concerns.

JHotDraw: JHotDraw is a well known open source GUI framework for drawing technical and structured graphics[10]. We have selected JHotDraw as LDA is observed to find a good set of concerns.

For software datasets, only the textual content (with-

[6]using inference as in (Williamson et al., 2010)
[7]nips.cc/Conferences/2005
[8]www.americanrhetoric.com/barackobamaspeeches.htm
[9]en.wikipedia.org/wiki/Berkeley_DB
[10]www.jhotdraw.org/

Table 1. Comparison on perplexity (top) and topic coherence (bottom). STM achieves lowest perplexity with good coherence.

Held-out data Perplexity				
Dataset	HDP	MG-LDA	FTM	STM
BerkeleyDB	182	127	80	**60**
JHotDraw	131	156	93	**81**
NIPS-05	941	2107	413	**402**
Obama-speech	3591	4721	901	**582**
Average	1211	1778	372	**281**

Average topic coherence				
Dataset	HDP	MG-LDA	FTM	STM
BerkeleyDB	-58.6	-49.6	**-20.3**	-27.9
JHotDraw	-80.9	-94.2	-37.9	**-28.2**
NIPS-05	-78.1	-43.7	-45.1	**-37.7**
Obama-speech	-72.4	-53.2	-67.2	**-52.5**
Average	-72.5	-59.9	-42.6	**-36.6**

Table 2. Comparison on average recall and average topic coherence considering only the gold standard topics with DoS greater than 0.2.

	HDP	MG-LDA	FTM	STM
BerkeleyDB				
Coherence	-48.48	-42.89	-40.11	**-21.99**
Recall	0.42	0.59	0.68	**0.94**
JHotDraw				
Coherence	**-36.64**	-52.13	-43.26	-46.17
Recall	0.31	0.16	0.42	**0.97**
NIPS-05				
Coherence	-40.23	-38.55	-37.38	**-34.84**
Recall	0.41	0.49	0.56	**0.79**
Obama-speech				
Coherence	-32.65	**-22.03**	-53.15	-40.6
Recall	0.26	0.27	0.58	**0.95**

Table 3. Fraction of subtle topics(DoS $\geq$ 0.5) detected(recall $\geq$ 0.75) by all the models.

	HDP	MG-LDA	FTM	STM
BerkeleyDB	0	0.25	0.38	**1.0**
JHotDraw	0	0	0.29	**0.86**
NIPS-05	0	0	0.07	**0.69**
Obama-speech	0.14	0.07	0.28	**0.93**

out programming syntax) of the '.java' files(no documentation etc) used as input. Each statement has been treated as a sentence and Java key-words[11] are removed but common java library names are retained. Tokens like StringCopy have been split into two words String and Copy based on the position of a capital face inside a token. For all datasets, we have removed standard English stop words, digits, sentences smaller than 20 characters and words smaller than 3 characters. We converted capital faces to small faces. In case of NIPS dataset, we have used most frequent 5000 words and in other cases we have used full vocabulary.

We have used the parameters $\alpha, \eta, \mu, \tau_j, \iota_j$ as 1 and δ as 100. We have run all the models for 2000 iterations(we found it sufficient for all models to converge in terms of log-likelihood), and used the truncation parameter K as 100(adequate considering the size of our datasets).

5.4. Evaluation on perplexity & coherence

We have randomly picked one-third of the datasets as held-out datasets and used the standard definition of perplexity as can be found in (Blei et al., 2003). Lower value in perplexity means that the model fits the dataset better. By approximating the user experience of topic quality on W top words of a topic topic coherence (TC) can be measured as: $TC(W) = \sum_i \sum_{j<i} \log \frac{D(w_i,w_j)+\epsilon}{D(w_j)}$ where $D(w)$ is the document frequency of any word w, and $D(w_i, w_j)$ is the document frequency of w_i and w_j together(Mimno et al., 2011). ϵ is a small constant to avoid log zero. Values closer to zero indicates better coherence. We have used top 5 words to compute coherence of a topic.

Table 1 contains results on perplexity and average topic coherence for all the datasets. We observe that STM is a better model than all others in terms held-out data perplexity and coherence(in most of the cases). Note that, the ability of STM to detect subtle topics lies in splitting the co-occurrence domain, however this brings in mild difficulty for normal topics to be learnt. Hence, coherence may suffer little bit which is observed in Table 1 too.

5.5. Evaluation in detecting subtle topics

Measure of subtlety: We define degree of subtlety of topic k as $DoS(k) = \prod_{d=1}^{D}(1 - p_{dk})$, where $p_{dk} = \frac{\sum_{w \in \mathcal{K}_k} \sum_{i=1}^{S_d} I[w \in S_{di}]}{|\mathcal{K}_k| S_d}$. $\mathcal{K}_k$ is the set of keywords describing topic k. $I[w \in S_{di}]$ is 1 if word w is present in sentence i of document d. Note that, $0 \leq DoS \leq 1$. The value of DoS increases if a rare word is included into $\mathcal{K}_k$ and it decreases if a frequent word is inserted.

Gold standard: In order to compare performance on subtle topics, we hand-picked some topics from each corpus so that their DoS is greater than 0.2 which we

[11] en.wikipedia.org/wiki/List_of_Java_keywords

Table 4. Five example subtle topics (DoS $\geq$ 0.5) detected(recall $\geq$ 0.75) by STM. Among them, '*' marked are also detected by MG-LDA and '!' marked are also detected by FTM. HDP could not detect any of them.

BerkeleyDB	JHotDraw	NIPS	Obama speech
transaction, checkpoint, recovery	nano, xmldom	chip, processing, architecture, circuit	cyber, security, internet
stats, count	roundrect	cochlear, cochlea	school, students, college
*checksum, validate, errors	!rendering	topic, model, topics, dirichlet	carnegie, mellon, technology
!trace, level, info, config	zoom, factor	walk, walks, steering, robot	deficit, cuts, budget
verify, config, keys	collection, family, families	video, texture, resolution, image	regulations, infrastructure, employees

consider as a reasonably challenging degree of subtlety. For comparison we however considered only those hand-picked topics for which a least 75% of the keywords are retrieved among top 5 words by at least one method in comparison. We compute recall of each topic considering top five words with $\mathcal{K}$ of each gold standard topic. A topic is said to be a match for the gold-standard topic which has the highest recall, if recall is less than 0.75, then we say the topic is not detected by the model. The reason of keeping threshold of recall high is that some keywords may be popular and part of normal topics. Hence, detecting such a keyword alone does not signify detecting the subtle topic in consideration.

Following the above approach, we hand picked 11, 10, 21, and 16 topics respectively from BerkeleyDB, JHotDraw, NIPS and Obama-speech datasets. For each gold standard topic we consider the recall and coherence of the best matched topic for each model and then we average over all the gold standard topics corresponding to each dataset and report at Table 2. Table 3 contains the result on fraction of subtle topics detected by all the models. A complete list can be found in the supplementary material. In Table 4, we provide five example subtle topics from each dataset which are detected by STM, but other models hardly detect them.

5.6. Discussion

Subtle topics in many cases consist of rare words. For example, the "*cochlea*" topic is subtle due to rareness of its keywords and is detected only by STM. In certain cases, some keywords may not be rare but it is the combination of the words that makes the topic subtle to detect. For example, in Berkeley-DB the topic "*trace, level, info, config*" consists of four words of which *trace* is not a rare word. The topic as a whole signifies the ability to configure trace levels, and manifests in a very localized fashion in the Tracer class and hence can be detected by FTM, but not by HDP or MG-LDA. On the other hand the cross-cutting concern "*checksum, errors, validate*" appears subtly in individual files but is diffused widely across the corpus and hence it can be detected by MG-LDA, but HDP and FTM fail. Not only STM detects all of these topics, but it also succeeds in detecting the more interesting cross-cutting concern, "*verify*", which manifests subtly in every file and therefore eludes HDP, FTM and MG-LDA. Another example of an important yet subtle topic detected only by STM is *"cyber security"* on the internet. This is an important topic that indicates policies or priorities of president Obama.

6. Conclusion

The utility of a topic is not necessarily linked to its prevalence in a corpus. When topic models are used for automatically discovering concerns, the inability to discover important or interesting concerns just because they are subtly manifested in the source code can be a critical drawback. In this paper we propose a novel model, namely STM to address this problem for the first time in the literature. STM, by using multiple distributions over topics per document is observed to effectively discover subtle topics where state of art models fail. This is a promising result for advancing the state-of-the-art in the difficult problem of automatic concern discovery in software engineering. On empirical evaluation, we find STM to out perform state of art models not only in case of subtle topics but also in general with low perplexity on unseen data, and good topic coherence.

Acknowledgments

We are thankful to all the reviewers for their valuable comments. The authors MKD and CB were partially supported by DST grant(DST/ECA/CB/1101).

References

Apel, Sven, Kastner, Christian, and Lengauer, Christian. FEATUREHOUSE: Language-Independent, Automated Software Composition. In *Proceedings of the 31st International Conference on Software Engineering(ICSE)*, pp. 221–231, 2009.

Baldi, Pierre F., Lopes, Cristina V., Linstead, Erik J., and Bajracharya, Sushil K. A theory of aspects as latent topics. In *Proceedings of the 23rd ACM SIGPLAN conference on Object-oriented programming systems languages and applications(OOPSLA)*, pp. 543–562, 2008.

Blei, David M., Ng, Andrew Y., and Jordan, Michael I. Latent Dirichlet Allocation. *The Journal of Machine Learning Research(JMLR)*, 3:993–1022, 2003.

Doshi-Velez, Finale, Miller, Kurt T., Gael, Jurgen Van, and Teh, Yee Whye. Variational Inference for the Indian Buffet Process. In *Proceedings of the Intl. Conf. on Artificial Intelligence and Statistics(AISTATS)*, pp. 137–144, 2009.

Eaddy, M., Aho, A. V., Antoniol, G., and Gueheneuc, Y. G. CERBERUS: Tracing Requirements to Source Code Using Information Retrieval, Dynamic Analysis, and Program Analysis. In *International Conference on Program Comprehension In Program Comprehension(ICPC)*, pp. 53–62, 2008.

Globerson, A., Chechik, G., Pereira, F., and Tishby, N. Euclidean Embedding of Co-occurrence Data. *The Journal of Machine Learning Research(JMLR)*, 8: 2265–2295, 2007.

Griffiths, T. L., Ghahramani, Z., and Sollich, Peter. Bayesian Nonparametric Latent Feature Models. In *Bayesian Statistics*, pp. 201–225, 2007.

Ishwaran, H. and James, L. F. Gibbs Sampling Methods for Stick-Breaking Priors. *Journal of the American Statistical Association*, 96:161–173, 2001.

Kastner, Christian, Apel, Sven, and Batory, Don. A Case Study Implementing Features Using AspectJ. In *Proceedings of the 11th International Software Product Line Conference(SPLC)*, pp. 223–232, 2007.

Marin, Marius, Deursen, Arie Van, and Moonen, Leon. Identifying crosscutting concerns using fan-in analysis. *ACM Transactions on Software Engineering and Methodology(TOSEM)*, 17, 2007.

Mimno, David, Wallach, Hanna, Talley, Edmund, Leenders, Miriam, and McCallum, Andrew. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing(EMNLP)*, pp. 262–272, 2011.

Revelle, Meghan, Dit, Bogdan, and Poshyvanyk, Denys. Using data fusion and web mining to support feature location in software. In *Proceedings of the 2010 IEEE 18th International Conference on Program Comprehension(ICPC)*, pp. 14–23, 2010.

Robillard, M. P. Topology Analysis of Software Dependencies. *ACM Transactions on Software Engineering and Methodology(TOSEM)*, (4), 2008.

Savage, T., Revelle, M., and Poshyvanyk, D. FLAT^3: Feature Location and Textual Tracing Tool. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering(ICSE) - Volume 2*, pp. 255–258, 2010a.

Savage, Trevor, Dit, Bogdan, Gethers, Malcom, and Poshyvanyk, Denys. TopicXP: Exploring topics in source code using Latent Dirichlet Allocation. In *Proceedings of the IEEE International Conference on Software Maintenance(ICSM)*, pp. 1–6, 2010b.

Teh, Y., Jordan, M. I., and Beal, M. Hierarchical Dirichlet processes. *Journal of American Statistical Association*, pp. 1566–1581, 2007.

Titov, Ivan and McDonald, Ryan. Modeling Online Reviews with Multi-grain Topic Models. In *Proceedings of the 17th International Conference on World Wide Web(WWW)*, pp. 111–120, 2008.

Wen, Bo and Boahen, Kwabena. Active Bidirectional Coupling in a Cochlear Chip. In *Advances in Neural Information Processing Systems(NIPS)*, 2005.

Williamson, Sinead, Wang, Chong, Heller, Katherine A., and Blei, David M. The IBP Compound Dirichlet Process and its Application to Focused Topic Modeling. In *Proceedings of the 27th International Conference on Machine Learning(ICML)*, 2010.

Wong, T. T. Generalized Dirichlet distribution in Bayesian analysis. *Applied Mathematics and Computation*, 97:165–181, 1998.