A Computationally Efficient Algorithm for Learning Topical Collocation Models

Zhendong Zhao[1], Lan Du[1], Benjamin Börschinger[1,2], John K Pate[1], Massimiliano Ciaramita[2], Mark Steedman[3] and Mark Johnson[1]

[1] Department of Computing, Macquarie University, Australia
[2] Google, Zurich, Switzerland
[3] School of Informatics, University of Edinburgh, Scotland

Abstract

Most existing topic models make the *bag-of-words* assumption that words are generated independently, and so ignore potentially useful information about word order. Previous attempts to use *collocations* (short sequences of adjacent words) in topic models have either relied on a pipeline approach, restricted attention to bigrams, or resulted in models whose inference does not scale to large corpora. This paper studies how to simultaneously learn both collocations and their topic assignments. We present an efficient reformulation of the Adaptor Grammar-based topical collocation model (AG-colloc) (Johnson, 2010), and develop a point-wise sampling algorithm for posterior inference in this new formulation. We further improve the efficiency of the sampling algorithm by exploiting sparsity and parallelising inference. Experimental results derived in text classification, information retrieval and human evaluation tasks across a range of datasets show that this reformulation scales to hundreds of thousands of documents while maintaining the good performance of the AG-colloc model.

1 Introduction

Probabilistic topic models like Latent Dirichlet Allocation (LDA) (Blei et al., 2003) are commonly used to study the meaning of text by identifying a set of latent topics from a collection of documents and assigning each word in these documents to one of the latent topics. A document is modelled as a mixture of latent topics, and each topic is a distribution over a finite vocabulary of words. It is common for topic models to treat documents as bags-of-words, ignoring any internal structure. While this simplifies posterior inference, it also ignores the information encoded in, for example, syntactic relationships (Boyd-Graber and Blei, 2009), word order (Wallach, 2006) and the topic structure of documents (Du et al., 2013). Here we are interested in topic models that capture dependencies between adjacent words in a topic dependent way. For example, the phrase "white house" can be interpreted compositionally in a real-estate context, but not in a political context.

Several extensions of LDA have been proposed that assign topics not only to individual words but also to multi-word phrases, which we call *topical collocations*. However, as we will discuss in section 2, most of those extensions either rely on a pre-processing step to identify potential collocations (e.g., bigrams and trigrams) or limit attention to bigram dependencies. We want a model that can jointly learn collocations of arbitrary length and their corresponding topic assignments from a large collection of documents. The AG-colloc model (Johnson, 2010) does exactly this. However, because the model is formulated within the Adaptor Grammar framework (Johnson et al., 2007), the time complexity of its inference algorithm is cubic in the length of each text fragment, and so it is not feasible to apply the AG-colloc model to large collections of text documents.

In this paper we show how to reformulate the AG-colloc model so it is no longer relies on a general Adaptor Grammar inference procedure. The new formulation facilitates more efficient inference by extending ideas developed for Bayesian word segmentation (Goldwater et al., 2009). We adapt a point-wise sampling algorithm from Bayesian word segmentation, which has also been used in Du et al. (2013), to simultaneously sample collocation boundaries and collocation topic assignments. This algorithm retains the good performance of the AG-colloc model in document classification and information retrieval

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics
and the 7th International Joint Conference on Natural Language Processing, pages 1460–1469,
Beijing, China, July 26-31, 2015.

tasks. By exploiting the sparse structure of both collocation and topic distributions, using techniques inspired by Yao et al. (2009), our new inference algorithm produces a remarkable speedup in running time and allows our reformulation to scale to a large number of documents. This algorithm can also be easily parallelised to take advantage of multiple cores by combining the ideas of the distributed LDA model (Newman et al., 2009). Thus, the contribution of this paper is three-fold: 1) a novel reformulation of the AG-colloc model, 2) an easily parallelisable and fast point-wise sampling algorithm exploiting sparsity and 3) systematic experiments with both qualitative and quantitative analysis.

The structure of the paper is as follows. In Section 2 we briefly discuss prior work on learning topical collocations. We then present our reformulation of the AG-colloc model in Section 3. Section 4 derives a point-wise Gibbs sampler for the model and shows how this sampler can take advantage of sparsity and be parallelised across multiple cores. Experimental results are reported in Section 5. Section 6 concludes this paper and discusses future work.

2 Related Work

There are two main approaches to incorporating topical collocations in LDA: 1) pipeline approaches that use a pre-processing step prior to LDA, and 2) extensions to LDA, which modify the generative process. In this section we discuss prior work that falls into these two categories and their limitations.

Pipeline Approaches (Lau et al., 2013), denoted here by PA, involve two steps. The first step identifies a set of bigrams that are potentially relevant collocations from documents by using simple heuristics for learning collocations, e.g., the Student's t-test method of Banerjee and Pedersen (2003). For each identified bigram "w_1 w_2", a new pseudo word "$\mathrm{w}_1\_\mathrm{w}_2$" is added to the vocabulary and the documents are re-tokenised to treat every instance of this bigram as a new token. LDA is then applied directly to the modified corpus without any changes to the model. While Lau et al. demonstrated that this two-step approach improves performance on a document classification task, it is limited in two ways. First, it can identify only collocations of a fixed length (i.e., bigrams). Second, the pre-processing step that identifies collocation candidates has no access to contextual cues (e.g. the topic of the context in which a bigram occurs),

A variety of extensions to the LDA model have been proposed to address this second shortcoming. Most extensions add some ability to capture word-to-word dependencies directly into the underlying generative process. For example, Wallach (2006) incorporates a hierarchical Dirichlet language model (MacKay and Peto, 1995), enabling her model to automatically cluster function words together. The model proposed by Griffiths et al. (2004) combines a hidden Markov model with LDA, using the former to model syntax and the latter to model semantics.

The LDA collocation model (LDACOL) (Griffiths et al., 2007) infers both the per-topic word distribution in the standard LDA model and, for each word in the vocabulary, a distribution over the words that follow it. The generative process of the LDACOL model allows words in a document to be generated in two ways. A word is generated either by drawing it directly from a per-topic word distribution corresponding to its topic as in LDA, or by drawing it from the word distribution associated with its preceding word w. The two alternatives are controlled by a set of Bernoulli random variables associated with individual words. Sequences of words generated from their predecessors constitute topical collocations.

Wang et al. (2007) extended the LDACOL model to generate the second word of a collocation from a distribution that conditions on not only the first word but also the first word's topic assignment, proposing the topical N-gram (TNG) model. In other words, whereas LDACOL only adds a distribution for every word-type to LDA, TNG adds a distribution for every possible word-topic pair. Wang et al. found that this modification allowed TNG to outperform LDACOL on a standard information retrieval task. However, both LDACOL and TNG do not require words within a sequence to share the same topic, which can result in semantically incoherent collocations.

Subsquent models have sought to encourage topically coherent collocations, including Phrase-Discovering LDA (Lindsey et al., 2012), the time-based topical n-gram model (Jameel and Lam, 2013a) and the n-gram Hierarchical Dirichlet Process (HDP) model (Jameel and Lam, 2013b). Phrase-Discovering LDA is a non-parametric extension of TNG inspired by Bayesian N-gram models Teh (2006) that incorporate a Pitman-Yor Process prior. The n-gram HDP is a nonparametric extension of LDA-colloc, putting an HDP prior on the per-document topic distribution. Both of these non-parametric extensions use the Chinese Franchise representation for posterior inference.

Our work here is based on the AG-colloc model proposed by Johnson (2010). He showed how Adaptor Grammars can generalise LDA to learn topical collocations of unbounded length while jointly identifying the topics that occur in each document. Unfortunately, because the Adaptor Grammar inference algorithm uses Probabilistic Context-Free Grammar (PCFG) parsing as a subroutine, the time complexity of inference is cubic in the length of individual text fragments. In order to improve the efficiency of the AG-colloc model, we re-express it using ideas from Bayesian word segmentation models. This allows us to develop an efficient inference algorithm for the AG-colloc model that scales to large corpora. Finally, we evaluate our model in terms of classification, information retrieval, and topic intrusion detection tasks; to our knowledge, we are the first to evaluate topical collocation models along all the three dimensions.

3 Topical Collocation Model

In this section we present our reformulation of the AG-colloc model, which we call the Topical Collocation Model (TCM) to emphasise that we are not using a grammar-based formulation. We start with the Unigram word segmentation model and Adaptor Grammar model of topical collocations, and then present our reformulation.

Goldwater et al. (2009) introduced a Bayesian model for word segmentation known as the Unigram model. This model is based on the Dirichet Process (DP) and assumes the following generative process for a sequence of words.

$$G \sim DP(\alpha_0, P_0), \qquad w_i \mid G \sim G$$

Here, P_0 is some distribution over the countably infinite set of all possible word forms (which are in turn sequences of a finite number of characters), and G is a draw from a Dirichlet Process. Inference is usually performed under a collapsed model in which G is integrated out, giving rise to a Chinese Restaurant Process (CRP) representation. The CRP is defined by the following predictive probability of w_i given $w_{1:i-1}$:

$$p(w_i = l | w_{1:i-1}) = \frac{n_l}{i-1+\alpha_0} + \frac{\alpha_0 P_0(l)}{i-1+\alpha_0},$$

where n_l is the number of times word form l appears in the first $n-1$ words.

During inference, the words are not known, and the model observes only a sequence of characters. Goldwater et al. (2009) derived a linear time Gibbs sampler that samples from the posterior distribution over possible segmentations of a given corpus according to the model. Their key insight is that sampling can be performed over a vector of Boolean boundary indicator variables – not included in the original description of the model – that indicates which adjacent characters are separated by a word boundary. We will show how this idea can be generalised to yield an inference algorithm for the AG-colloc model.

Adaptor Grammars (Johnson et al., 2007) are a generalisation of PCFGs. In a PCFG, a non-terminal A is expanded by selecting a rule $A \rightarrow \beta$ with probability $P(\beta|A)$, where β is a sequence of terminal and non-terminal node labels. Because the rules are selected independently, PCFGs introduce strong conditional independence assumptions. In an Adaptor Grammar, some of the non-terminal labels are *adapted*. These nodes can be expanded either by selecting a rule, as in PCFGs, or by retrieving an entire subtree from a Dirichlet Process cache specific to that node's non-terminal label,[1] breaking the conditional independence assumptions and capturing longer-range statistical relationships.

The AG-colloc model can be concisely expressed using context free grammar rule schemata, where adapted non-terminals are underlined:

$$\begin{aligned} \text{Top} &\rightarrow \text{Doc}_m \\ \text{Doc}_m &\rightarrow_{-m} \mid \text{Doc}_m\ \underline{\text{Topic}_i} \\ \underline{\text{Topic}_i} &\rightarrow \text{Word}^+ \end{aligned}$$

Here m ranges over the documents, i ranges over topics, "|" separates possible expansions, and "+" means "one or more". As in LDA, each document is defined as a mixture of K topics with the mixture probabilities corresponding to the probabili-

[1] Strictly speaking, Adaptor Grammars are defined using the Pitman-Yor process. In this paper we restrict ourselves to considering the Dirichlet Process which is a special case of the PYP where the discount parameter is set to 0. For more details, refer to Johnson et al. (2007) and Johnson (2010).

ties of the different expansions of Doc_m. However, the topic distributions are modelled using an adapted non-terminal $\underline{\text{Topic}_i}$. This means that there is an infinite number of rules expanding $\underline{\text{Topic}_i}$, one for every possible sequence over the finite vocabulary of words. $\underline{\text{Topic}_i}$ non-terminals cache sequences of words, just as G caches sequences of characters in the Unigram model.

The base distribution of the AG-colloc model is a geometric distribution over sequences of a finite vocabulary of words: $P_0(c = (w_1, \ldots, w_M)) = p_\#(1-p_\#)^{M-1} \prod_{j=1}^{M} P_w(w_j)$, where $P_w(\cdot)$ is the uniform distribution over the finite set of words. This is the same base distribution used by Goldwater et al. (2009), except characters have been replaced by words. $p_\#$ is the probability of seeing the end of a collocation, and so controls the length of collocations. With this, we can re-express the AG-colloc model as a slight modification of the Unigram model:

1. For each topic $k, 1 \leq k \leq K$, $\phi_k \sim \text{DP}(\alpha_0, P_0)$
2. For each document $d, 1 \leq d \leq D$
   (a) Draw a topic distribution $\boldsymbol{\theta}_d | \alpha \sim \text{Dirichlet}_K(\alpha)$
   (b) For each collocation $c_{d,n}$ in document $d, 1 \leq n \leq N_d$
      i. Draw a topic assignment: $z_{d,n} \mid \boldsymbol{\theta}_d \sim \text{Discrete}(\boldsymbol{\theta}_d)$
      ii. Draw a collocation: $c_{d,n} \mid z_{d,n}, \boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K \sim \boldsymbol{\phi}_{z_{d,n}}$

where the length of a collocation $c_{d,n}$ is greater than or equal to 1, i.e., $|c_{d,n}| \geq 1$. Unlike previous models, the TCM associates each topic with a Unigram model over topical collocations. Therefore, the TCM learns different vocabularies for different topics.[2]

4 Posterior Inference

We develop an efficient point-wise sampling algorithm that can jointly sample collocations and their topics. The observed data consists of a sequence of word tokens which are grouped into D documents. We sample from the posterior distribution over segmentations of documents into collocations, and assignments of topics to collocations. Let each document d be a sequence of N_d words $w_{d,1}, \ldots, w_{d,N_d}$. We introduce a set of auxiliary random variables $b_{d,1}, \ldots, b_{d,N_d}$. The value of $b_{d,j}$ indicates whether there is a collocation boundary between $w_{d,j}$ and $w_{d,j+1}$, and, if there is, the topic of the collocation to the left of the boundary. If there is no boundary then $b_{d,j} = 0$. Otherwise, there is a collocation to the left of the boundary consisting of the words $w_{d,l+1}, \ldots, w_{d,j}$ where $l = \max\{i \mid 1 \leq i \leq j-1 \wedge b_{d,i} \neq 0\}$, and $b_{d,j} = k$ $(1 \leq k \leq K)$ is the topic of the collocation. Note that b_{d,N_d} must not be 0 as the end of a document is always a collocation boundary.

For example, consider the document consisting of the words "the white house." We use the K+1-valued variables b_1, b_2 (after 'the' and 'white') and the K-valued variable b_3 (after 'house') to describe every possible segmentation of this document into topical collocations.[3] If there are K topics and N words, there are $(K+1)^{N-1}K$ possible topical segmentations. To illustrate, see how each of the following triples (b_1, b_2, b_3) encodes a different analysis of "the white house" into bracketed collocations and subscripted topic numbers:

- $(0,0,1)$: (the white house)$_1$
- $(1,0,2)$: (the)$_1$ (white house)$_2$
- $(2,1,1)$: (the)$_2$ (white)$_1$ (house)$_1$

The next section elaborates the Gibbs sampler over these K+1 boundary variables.

4.1 A Point-wise Gibbs Sampler for the TCM

We consider a collapsed version of the TCM in which the document-specific topic mixtures $\boldsymbol{\theta}_{1:D}$ and the K non-parametric topic distributions $\boldsymbol{\phi}_{1:K}$ are integrated out. We introduce the sampling equations using a concrete example, considering again the toy document, "the white house."

Let the sampler start in state $b_1 = b_2 = 0$, $b_3 = z_0, 1 \leq z_0 \leq K$. This corresponds to the analysis

$$\underbrace{(\text{the}_0 \ \text{white}_0 \ \text{house}_{z_0})}_{c_0}.$$

This analysis consists of a single collocation c_0 which spans the entire document and is assigned to topic z_0. For simplicity, we will not show how to model document boundaries.

If we resample b_1, we have to consider two different hypotheses, i.e., putting or not putting a collocation boundary at b_1. The analysis corresponding to not putting a boundary is the one we just

[2] In the TCM, the vocabulary differs from topic to topic. Given a sequence of adjacent words, it is hard to tell if it is a collocation without knowing the topic of its context. Therefore, the Pointwise Mutual Information (PMI) (Newman et al., 2010) and its variant (Lau et al., 2014) are not applicable to our TCM in evaluation.

[3] A similar strategy of using K-valued rather than boolean boundary variables in Gibbs sampling was used in Börschinger et al. (2013) and Du et al. (2014).

saw. Putting a boundary corresponds to a new segmentation,

$$\underbrace{(\text{the}_{z_1})}_{c_1}\underbrace{(\text{white}_0\ \text{house}_{z_2})}_{c_2}.$$

We need to consider the K possible topics for c_1, for each of which we calculate the probability as follows. If $b_1 = 0$ (i.e., there is no collocation boundary after "the") we have

$$p(z_0, c_0|\boldsymbol{\mu}) = p(z_0|\alpha)p(c_0|\alpha_0, P_0, z_0)\,, \quad (1)$$

where $\boldsymbol{\mu} = \{\alpha, \alpha_0, P_0\}$. $p(c_0|\alpha_0, P_0, z_0)$ is the probability of generating collocation c_0 from topic z_0 with a CRP, i.e.,

$$p(c_0|\alpha_0, P_0, z_0) = \frac{n_{z_0}^{-c_0} + \alpha_0 P_0(c_0)}{N_{z_0}^{-c_0} + \alpha_0}\,, \quad (2)$$

where $n_{z_0}^{-c_0}$ is the number of times that collocation c_0 was assigned to topic z_0 and $N_{z_0}^{-c_0}$ is the total number of collocations assigned to z_0. Both counts exclude the parts of the analysis that are affected by the boundary c_0. As in LDA,

$$p(z_0 = k|\alpha) = \frac{\hat{n}_k^{-c_0} + \alpha}{\sum_{k=1}^{K} \hat{n}_k^{-c_0} + K\alpha}\,, \quad (3)$$

where $\hat{n}_k^{-c_0}$ is the total number of collocations assigned to topic k in a document, again excluding the count for the parts of the document that are affected by the current boundary. For the hypothesis that $b_1 = z_1$ (with $1 \leq z_1 \leq K$), the full conditional to generate two adjacent collocations is

$$\begin{aligned} &p(z_1, z_2, c_1, c_2|\boldsymbol{\mu}) \propto \quad (4)\\ &\quad p(z_1|\alpha)p(c_1|\alpha_0, P_0, z_1)\\ &\quad p(z_2|\alpha, z_1)p(c_2|\alpha_0, P_0, c_1, z_1, z_2)\,, \end{aligned}$$

where $p(z_1|\alpha)$ and $p(c_1|\alpha_0, P_0, z_1)$ can be computed with Eqs (3) and (2), respectively. The remaining probabilities are computed as

$$\begin{aligned} &p(z_2 = k|\alpha, z_1) =\\ &\qquad \frac{\hat{n}_k^{-c_1,c_2} + \alpha + \mathbb{I}_{z_2=z_1}}{\sum_{k=1}^{K} \hat{n}_k^{-c_1,c_2} + K\alpha + 1}\,, \end{aligned} \quad (5)$$

$$\begin{aligned} &p(c_2|\alpha_0, P_0, c_1, z_1, z_2) =\\ &\qquad \frac{n_{z_2}^{-c_1,c_2} + \mathbb{I}_{z_1=z_2}\mathbb{I}_{c_1=c_2} + \alpha_0 P_0(c_2)}{\alpha_0 + N_{z_2}^{-c_1,c_2} + \mathbb{I}_{z_1=z_2}} \end{aligned} \quad (6)$$

where $\mathbb{I}_{x=y}$ is an indicator function that is equal to 1 if $x = y$ and 0 otherwise, $n_{z_2}^{-c_1,c_2}$ is the number of collocations c_2 assigned to topic z_2, and $N_{z_2}^{-c_1,c_2}$ is the total number of collocations assigned to topic z_2. Both counts exclude the current c_2, and also exclude c_1 if $z_1 = z_2$ and $c_1 = c_2$. Our sampler does random sweeps over all the boundary positions, and calculates the joint probability of the corresponding collocations and their topic assignment using Eqs (1) and (4) at each position.

4.2 Parallelised Sparse Sampling Algorithm

The word distributions and topic distributions in LDA are typically sparse, and Yao et al. (2009) proposed a 'sparseLDA' Gibbs sampler that takes advantage of this sparsity to substantially reduce running time. These two distributions are even sparser for the TCM than LDA, because collocations are less frequent than unigrams. Here we show how to modify our sampler to take advantage of sparsity. Sampling boundaries according the two probabilities shown Eqs (1) and (4) requires the generation of a random number x from a uniform distribution, $\mathcal{U}(0, \mathcal{P})$, where

$$\mathcal{P} = p(z_0, c_0) + \sum_{z_1=1}^{K} p(z_1, c_1)p(z_2, c_2|c_1, z_1)\,. \quad (7)$$

Here the first term corresponds to the case that there is no collocation boundary, and the summation corresponds to the case that there is a collocation boundary. Thus, if x is less than $P(z_0, c_0)$, there will be no boundary. Otherwise, we need to sample z_1 according to Eq (4).

The sampling algorithm requires calculation of Eq (7), even though the probability mass may be concentrated on just a few topics. We have observed in our experiments that the denominators of Eqs (5) and (6) are often quite large and the indicator functions usually turn out to be zero, so we approximate the two equations by removing the indicator functions. This approximation not only facilitates the computation of Eq (7), but also means that $p(z_2, c_2|c_1, z_1)$ no longer depends on z_1 and c_1. Thus, Eq (7) can be approximated as

$$\mathcal{P} \approx p(z_0, c_0) + p(z_2, c_2) \sum_{z_1=1}^{K} p(z_1, c_1)\,. \quad (8)$$

Now that $p(z_0, c_0)$ and $p(z_2, c_2)$ are both out of the summation; they can be pre-computed and cached.

To reduce the computational complexity of the summation term in Eq (8), we use the "buckets" method (Yao et al., 2009). We divide the summation term in $p(z_1, c_1)$ into three parts as follows, each of which corresponds to a bucket:

$$\begin{aligned} & p(z_1 = k, c_1) \\ = \; & \frac{\hat{n}_k^{-c_1,c_2} + \alpha}{\sum_{k=1}^{K} \hat{n}_k^{-c_1,c_2} + K\alpha} \frac{n_k^{-c_1,c_2} + \alpha_0 P_0(c_1)}{N_k^{-c_1,c_2} + \alpha_0} \\ \propto \; & \frac{\alpha_0 P_0(c_1)\alpha}{N_k^{-c_1,c_2} + \alpha_0} + \frac{\hat{n}_k^{-c_1,c_2}\alpha_0 P_0(c_1)}{N_k^{-c_1,c_2} + \alpha_0} \\ & + \frac{(\hat{n}_k^{-c_1,c_2} + \alpha) n_k^{-c_1,c_2}}{N_k^{-c_1,c_2} + \alpha_0} \end{aligned} \tag{9}$$

Then, the summation in Eq (8) is proportional to the sum of the following three equations:

$$s = \sum_{k=1}^{K} \frac{\alpha_0 P_0(c_1)\alpha}{N_k^{-c_1,c_2} + \alpha_0} \tag{10}$$

$$r = \sum_{k=1}^{K} \frac{\hat{n}_k^{-c_1,c_2}\alpha_0 P_0(c_1)}{N_k^{-c_1,c_2} + \alpha_0} \tag{11}$$

$$q = \sum_{k=1}^{K} \frac{(\hat{n}_k^{-c_1,c_2} + \alpha) n_k^{-c_1,c_2}}{N_k^{-c_1,c_2} + \alpha_0} \tag{12}$$

We can now use the sampling techniques used in the sparse-LDA model to sample z_1. Firstly, sample $U \sim \mathcal{U}(0, s + r + q)$. If $U < s$ we have hit bucket s. In this case, we need to compute the probability for each possible topic. If $s < x < (s + r)$ we have hit the second bucket r. In this case, we compute probabilities only for topics such that $\hat{n}_k^{-c_1,c_2} \neq 0$. If $x > (s + r)$ we have hit bucket q, which is the "topic collection" bucket, and we need only consider topics such that $n_k^{-c_1,c_2} \neq 0$. Although we use an approximation in computing the full conditionals, experimental results have shown that our TCM is as accurate as the original AG-colloc model, see Section 5.

Our sparse sampling algorithm can be easily parallelised with the same multi-threading strategy used by Newman et al. (2009) in their distributed LDA (AD-LDA). In AD-LDA, documents are distributed evenly across P processors, each of which also has a copy of the word-topic count matrix. Gibbs updates are performed simultaneously on each of the P processors. At the end of each Gibbs iteration, the P copies of the word-topic count matrices are collected and summed into the global word-topic count matrix.

In the TCM, collocations in each topic are generated from a CRP. Hence, distributing the word-topic count matrix in AD-LDA now corresponds to distributing a set of Chinese restaurants in the parallelised TCM. The challenge is how to merge the Chinese Restaurant copies from the P processors into a single global restaurant for each topic, similar to the merging problem in Du et al. (2013). However, Eqs (2) and (6) show that the statistics that need to be collected are the number of collocations generated for each topic. The number of tables in a restaurant does not matter.[4] Therefore, we can adapt the summation technique used in AD-LDA.

We further observed that if P is large, using a single processor to perform the summation operation could result in a large overhead. The summation step could be even costlier in TCM than in LDA, since the number of distinct collocations is much larger than the number of distinct words. Thus we also parallelise the summation step using all the processors that are free in this step.

5 Experimental Results

In this section we evaluate the effectiveness and efficiency of our Topical Collocation Model (TCM) on different tasks, i.e., a document classification task, an information retrieval task and a topic intrusion detection task. All the empirical results show that our TCM performs as well as the AG-colloc model and outperforms other collocation models (i.e., LDACOL (Griffiths et al., 2007), TNG (Wang et al., 2007), PA (Lau et al., 2013)). The TCM also runs much faster than the other models. We also compared the TCM with the Mallet implementation of AD-LDA (Newman et al., 2009), denoted by Mallet-LDA, for completeness. Following Griffiths et al. (2007), we used punctuation and Mallet's stop words to split the documents into subsequences of word tokens, then removed those punctuation and stop words from the input. All experiments were run on a cluster with 80 Xeon E7-4850 processors (2.0GHz) and 96 GB memory.

5.1 Classification Evaluation

In the classification task, we used three datasets: the movie review dataset (Pang and Lee, 2012) (**MReviews**), the 20 Newsgroups dataset, and the Reuters-21578 dataset. The movie review dataset includes 1,000 positive and 1,000 negative reviews. The 20 Newsgroups dataset is organised

[4] The number of tables is used only when sampling the concentration parameters, α_0, see Blunsom et al. (2009).

Task	Classification	IR
Dataset	MReview	SJMN-2k
Mallet-LDA	71.30	18.85
LDACOL	71.75	19.03
TNG	71.40	19.06
PA	72.74	19.16
AG-colloc	***73.15***	***19.37***
Non-sparse TCM	**73.14**	**19.30**
Sparse TCM	**73.13**	**19.31**

Table 1: Comparison of all models in the classification task (accuracy in %) and the information retrieval task (MAP scores in %) on small corpora. Bold face indicates scores not significantly different from the best score (in italics) according to a Wilcoxon signed rank test ($p < 0.05$).

	Mallet-LDA	PA	TCM
Politics	**89.1**	***89.2***	***89.2***
Comp	86.3	87.4	*87.9*
Sci	92.0	93.2	*93.4*
Sports	91.6	91.7	*92.6*
Reuter-21578	97.3	**97.5**	***97.6***

Table 2: Classification accuracy (%) on larger datasets. Bold face indicates scores not significantly different from the best score (in italics) according to a Wilcoxon signed rank test ($p < 0.05$).

into 20 different categories according to different topics. We further partitioned the 20 newsgroups dataset into four subsets, denoted by **Comp**, **Sci**, **Sport**, and **Politics**. They have $4,891$, $3,952$, $1,993$, and $2,625$ documents respectively. We applied document classification to each subset. The **Reuters-21578** dataset has 21,578 Reuters news articles which are split into 10 categories.

The classification evaluation was carried out as follows. First, we ran each model on each dataset to derive point estimates of documents' topic distributions ($\boldsymbol{\theta}$), which were used as the only features in classification. We then randomly selected from each dataset 80% documents for training and 20% for testing. A Support Vector Machine (SVM) with a linear-kernel was used. We ran all models for 10,000 iterations with 50 topics on the movie review dataset and 100 on the other two. We set $\alpha = 1/K$ and $\beta = 0.02$ for Mallet-LDA, LDACOL, TNG and PA. We used the reported settings in Johnson (2010) for the AG-colloc model. For the TCM, we used $\alpha = 1/K$. The concentration parameter α_0 was initially set to 100 and resampled using approximated table counts (Blunsom et al., 2009).

	Mallet-LDA	PA	TCM
SJMN	20.7	20.9	***21.2***
AP	24.0	24.5	***24.8***

Table 3: Mean average Precision (MAP in %) scores in the information retrieval task. Scores in bold and italics are the significantly best MAP scores according to a Wilcoxon signed rank test ($p < 0.05$).

Since efficient inference is unavailable for LDACOL, TNG and AG-colloc, making it impractical to evaluate them on the large corpora, we compared our TCM with them only on the **MReviews** dataset. The first column of Table 1 shows the classification accuracy of those models. All the collocation models outperform Mallet-LDA. The AG-colloc model yields the highest classification accuracy, and our TCM with/without sparsity performs as well as the AG-colloc model according to the Wilcoxon signed rank test. The Pipeline Approach (PA) is always better than LDACOL and TNG. Therefore, in the following experiments we will focus on the comparison among our TCM, Mallet-LDA and PA.

Table 2 shows the classification accuracy of those three models on the larger datasets, i.e., the 20 Newsgroups dataset, and the Reuters-21578 dataset. The TCM outperforms both Mallet-LDA and PA on 3 out of 5 datasets, and performs equally well as PA on the **Politics** and **Reuter-21578** datasets according to a Wilcoxon signed rank test ($p < 0.05$).

5.2 Information Retrieval Evaluation

For the information retrieval task, we used the method presented by Wei and Croft (2006) and Wang et al. (2007) to calculate the probability of a query given a document. We used the San Jose Mercury News (**SJMN**) dataset and the **AP** News dataset from TREC. The former has 90,257 documents, the latter has 242,918 documents. Queries 51-150 were used. We ran all the models for 10,000 iteration with 100 topics. The other parameter settings were the same as those used in Section 5.1. Queries were tokenised using unigrams for Mallet-LDA and collocations for all collocation models.

Models	$p(w\|t)$	$p(t\|w)$
Mallet-LDA	71.9	73.2
PA	72.8	76.7
TCM	**73.2**	**79.7**

Table 4: The model precision (%) derived from the intrusion detection experiments.

On a small subset of the SJMN data, which contains 2,000 documents (**SJMN-2k**), we find again that TCM and AG-colloc perform equally well and outperform all other models (LDACOL, TNG, PA), as shown in the second column of Table 1. We further compare the TCM, Mallet-LDA and PA on the full **SJMN** dataset and the **AP** news dataset, as these models can run on large scale. Table 3 shows the mean average precision (MAP) scores. The TCM significantly outperforms both Mallet-LDA and the PA approach, and yields the highest MAP score.

5.3 Topic Coherence Evaluation

We ran a set of topic intrusion detection experiments (Chang et al., 2009) that provide a human evaluation of the coherence of the topics learnt by Mallet-LDA, PA and TCM on the **SJMN** dataset. This set of experiments was use to measure how well the inferred topics match human concepts. Each subject recruited from Amazon Mechanical Turk was presented with a randomly ordered list of 10 tokens (either words or collocations). The task of the subject was to identify the token which is semantically different from the others.

To generate the 10-token lists, we experimented with two different methods for selecting tokens (either words or collocations) most strongly associated with a topic t. The standard method chooses the tokens w that maximise $p(w|t)$. This method is biased toward high frequency tokens, since low-frequency tokens are unlikely to have a large $p(w|t)$. We also tried choosing words and collocations w that maximise $p(t|w)$. This method finds w that are unlikely to appear in any other topic except t, and is biased towards low frequency w. We reduce this low-frequency bias by using a smoothed estimate for $p(t|w)$ with a Dirichlet pseudo-count $\alpha = 5$.

An intruder token was randomly selected from a set of tokens that had low probability in the current topic but high probability in some other topic. We then randomly selected one of the 10 tokens to be replaced by the intruder token. We expect collocations to be more useful in lists that are constructed using $p(t|w)$ than lists constructed using $p(w|t)$. This is because $p(w|t)$ can be dominated by the frequency of w, but individual collocations are rare.

Dataset	MReview		SJMN-2k	
#Topic	100	800	100	800
AG-colloc	84.9	1305	37.5	692
Non-sparse TCM	13.8	233	6.6	85.7
Sparse TCM	0.28	0.35	0.14	0.2

Table 5: The average running time (in seconds) per iteration.

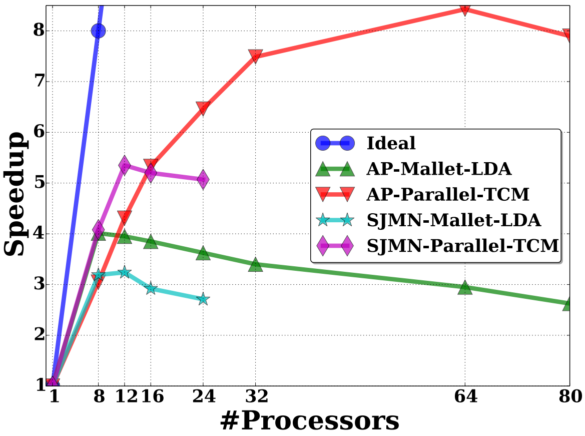

Figure 1: Plot of speedup in running time for the Mallet-LDA and our TCM.

The performance was measured by model precision (Chang et al., 2009), which measures the fraction of subjects that agreed with the model. Table 4 shows that our TCM outperforms both PA and Mallet-LDA under both ways of constructing the intrusion lists. As expected, the collocation models PA and TCM perform better with lists constructed according to $p(t|w)$ than lists constructed according to $p(w|t)$.

5.4 Efficiency of the TCM

In this section we study the efficiency of our TCM model in terms of running time. We first compare the efficiency of our TCM model with and without sparsity with the AG-colloc model on the **MReview** dataset and the **SJMN-2k** dataset. Table 5 shows the average running time per iteration for the two models. We used 100 and 800 topics. The TCM algorithm that does not exploit sparsity in sampling runs about 6 times faster than the AG-colloc model. Our sparse sampler runs even faster, and takes less than a second per iteration. Therefore, Tables 1 and 5 jointly show that our reformulation runs an order of magnitude faster than AG-colloc without losing performance, thereby making the AG-colloc model inference feasible at large scales.

We further studied the scalability of our sampling algorithm after parallelisation on the **SJMN** dataset and the **AP** news dataset. We fixed the number of topics to 100, and varied the number of processors from 1 to 24 for the SJMN dataset and from 1 to 80 for the AP dataset. The plots in Figure 1 show that our parallelised sampler achieved a remarkable speedup. We have also observed that there is a point at which using additional processors actually slows running time. This is common in parallel algorithms when communication and synchronisation take more time than the time saved by parallelisation. This slowdown occurs in the highly-optimized Mallet implementation of LDA with fewer cores than it does in our implementation. The speedup achieved by our TCM also shows the benefit of parallelising the summation step mentioned in Section 4.2.

6 Conclusion

In this paper we showed how to represent the AG-colloc model without using Adaptor Grammars, and how to adapt Gibbs sampling techniques from Bayesian word segmentation to perform posterior inference under the new representation. We further accelerated the sampling algorithm by taking advantage of the sparsity in the collocation count matrix. Experimental results derived in different tasks showed that 1) our new representation performs as well as the AG-colloc model and outperforms the other collocation models, 2) our point-wise sampling algorithm scales well to large corpora. There are several ways in which our model can be extended. For example, our algorithm could be further sped up by using the sampling techniques presented by Smola and Narayanamurthy (2010), Li et al. (2014) and Buntine and Mishra (2014). One can also consider using a hybrid of MCMC and variational inference as in Ke et al. (2014).

Acknowledgments

This research was supported by a Google award through the Natural Language Understanding Focused Program, and under the Australian Research Council's *Discovery Projects* funding scheme (project numbers DP110102506 and DP110102593).

References

Satanjeev Banerjee and Ted Pedersen. 2003. The design, implementation, and use of the ngram statistics package. In *Computational Linguistics and Intelligent Text Processing*, volume 2588, pages 370–381.

D.M. Blei, A.Y. Ng, and M.I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Phil Blunsom, Trevor Cohn, Sharon Goldwater, and Mark Johnson. 2009. A note on the implementation of hierarchical dirichlet processes. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 337–340.

Benjamin Börschinger, Mark Johnson, and Katherine Demuth. 2013. A joint model of word segmentation and phonological variation for english word-final /t/-deletion. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1508–1516, Sofia, Bulgaria.

Jordan L Boyd-Graber and David Blei. 2009. Syntactic topic models. In *Advances in Neural Information Processing Systems 21*, pages 185–192.

Wray L Buntine and Swapnil Mishra. 2014. Experiments with non-parametric topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 881–890.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems 22*, pages 288–296.

Lan Du, Wray Buntine, and Mark Johnson. 2013. Topic segmentation with a structured topic model. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200.

Lan Du, John Pate, and Mark Johnson. 2014. Topic models with topic ordering regularities for topic segmentation. In *Proceedings of the IEEE International Conference on Data Mining*, pages 803–808.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–53.

Thomas L Griffiths, Mark Steyvers, David M Blei, and Joshua B Tenenbaum. 2004. Integrating topics and syntax. In *Advances in neural information processing systems*, pages 537–544.

Thomas L. Griffiths, Mark Steyvers, and Joshua B. Tenenbaum. 2007. Topics in semantic representation. *Psychological Review*, 114(2):211–244.

Shoaib Jameel and Wai Lam. 2013a. An n-gram topic model for time-stamped documents. In *Proceedings of the 35th European Conference on Advances in Information Retrieval*, pages 292–304.

Shoaib Jameel and Wai Lam. 2013b. A nonparametric n-gram topic model with interpretable latent topics. In *Information Retrieval Technology*, pages 74–85.

M. Johnson, T.L. Griffiths, and S. Goldwater. 2007. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems 19*, pages 641–648.

Mark Johnson. 2010. Pcfgs, topic models, adaptor grammars and learning topical collocations and the structure of proper names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1157.

Zhai Ke, Boyd-Graber Jordan, and Cohen Shay B. 2014. Online adaptor grammars with hybrid inference. *Transactions of the Association of Computational Linguistics*, 2:465–476.

Jey Han Lau, Timothy Baldwin, and David Newman. 2013. On collocations and topic models. *ACM Transactions on Speech and Language Processing (TSLP)*, 10(3):10.

Jey Han Lau, David Newman, and Timothy Baldwin. 2014. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539.

Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. 2014. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 891–900.

Robert Lindsey, William Headden, and Michael Stipicevic. 2012. A phrase-discovering topic model using hierarchical Pitman-Yor processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 214–222.

David JC MacKay and Linda C Bauman Peto. 1995. A hierarchical Dirichlet language model. *Natural language engineering*, 1(3):289–308.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2009. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10:1801–1828.

David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. 2010. Evaluating topic models for digital libraries. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, pages 215–224.

Bo Pang and Lillian Lee. 2012. Cornell Movie Review Data.

Alexander Smola and Shravan Narayanamurthy. 2010. An architecture for parallel topic models. *Proc. VLDB Endow.*, 3(1-2):703–710.

Y. W. Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992.

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

Xuerui Wang, Andrew McCallum, and Xing Wei. 2007. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 697–702.

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*, pages 178–185.

Limin Yao, David Mimno, and Andrew McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 937–946.