Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval

Xuerui Wang, Andrew McCallum, Xing Wei
University of Massachusetts
140 Governors Dr, Amherst, MA 01003
{xuerui, mccallum, xwei}@cs.umass.edu

Abstract

Most topic models, such as latent Dirichlet allocation, rely on the bag-of-words assumption. However, word order and phrases are often critical to capturing the meaning of text in many text mining tasks. This paper presents topical n-grams, a topic model that discovers topics as well as topical phrases. The probabilistic model generates words in their textual order by, for each word, first sampling a topic, then sampling its status as a unigram or bigram, and then sampling the word from a topic-specific unigram or bigram distribution. Thus our model can model "white house" as a special meaning phrase in the 'politics' topic, but not in the 'real estate' topic. Successive bigrams form longer phrases. We present experimental results showing meaningful phrases and more interpretable topics from the NIPS data and improved information retrieval performance on a TREC collection.

1 Introduction

Although the bag-of-words assumption is prevalent in document classification and topic models, the great majority of natural language processing methods represent word order, including n-gram language models for speech recognition, finite-state models for information extraction and context-free grammars for parsing. Word order is not only important for syntax, but also important for lexical meaning. A collocation is a phrase with meaning beyond the individual words. For example, the phrase "white house" carries a special meaning beyond the appearance of its individual words, whereas "yellow house" does not. Note, however, that whether or not a phrase is a collocation may depend on the topic context. In the context of a document about real estate, "white house" may not be a collocation.

N-gram phrases are fundamentally important in many areas of natural language processing and text mining, including parsing, machine translation and information retrieval. In general, phrases as the whole carry more information than the sum of its individual components, thus they are much more crucial in determining the topics of collections than individual words. Most topic models such as latent Dirichlet allocation (LDA) [2], however, assume that words are generated independently from each other, i.e., under the bag-of-words assumption. Adding phrases increases the model's complexity, but it could be useful in certain contexts. The possible over complicacy caused by introducing phrases makes these topic models completely ignore them. It is true that these models with the bag-of-words assumption have enjoyed a big success, and attracted a lot of interests from researchers with different backgrounds. We believe that a topic model considering phrases would be definitely more useful in certain applications.

Assume that we conduct topic analysis on a large collection of research papers. The acknowledgment sections of research papers have a distinctive vocabulary. Not surprisingly, we would end up with a particular topic on acknowledgment (or funding agencies) since many papers have an acknowledgment section that is not tightly coupled with the content of papers. One might therefore expect to find words such as "thank", "support" and "grant" in a single topic. One might be very confused, however, to find words like "health" and "science" in the same topic, unless they are presented in context: "National Institutes of Health" and "National Science Foundation".

Phrases often have specialized meaning, but not always. For instance, "neural networks" is considered a phrase because of its frequent use as a fixed expression. However, it specifies two distinct concepts: biological neural networks in neuroscience and artificial neural networks in modern usage. Without consulting the context in which the term is located, it is hard to determine its actual meaning. In many situations, topic is very useful to accurately capture the meaning. Furthermore, topic can play a role in phrase discovery. Considering learning English, a beginner usually has difficulty in telling "strong tea" from "powerful tea" [15], which

are both grammatically correct. The topic associated with "tea" might help to discover the misuse of "powerful".

In this paper, we propose a new topical n-gram (TNG) model that automatically determines unigram words and phrases based on context and assign mixture of topics to both individual words and n-gram phrases. The ability to form phrases only where appropriate is unique to our model, distinguishing it from the traditional collocation discovery methods discussed in Section 3, where a *discovered* phrase is always treated as a *collocation* regardless of the context (which would possibly make us incorrectly conclude that "white house" remains a phrase in a document about real estate). Thus, TNG is not only a topic model that uses phrases, but also help linguists discover meaningful phrases in right context, in a completely probabilistic manner. We show examples of extracted phrases and more interpretable topics on the NIPS data, and in a text mining application, we present better information retrieval performance on an ad-hoc retrieval task over a TREC collection.

2 N-gram based Topic Models

Before presenting our topical n-gram model, we first describe two related n-gram models. Notation used in this paper is listed in Table 1, and the graphical models are showed in Figure 1. For simplicity, all the models discussed in this section make the 1^{st} order Markov assumption, that is, they are actually bigram models. However, all the models have the ability to "model" higher order n-grams ($n > 2$) by concatenating consecutive bigrams.

2.1 Bigram Topic Model (BTM)

Recently, Wallach develops a bigram topic model [22] on the basis of the hierarchical Dirichlet language model [14], by incorporating the concept of topic into bigram models. This model is one solution for the "neural network" example in Section 1. We assume a dummy word w_0 existing at the beginning of each document. The graphical model presentation of this model is shown in Figure 1(a). The generative process of this model can be described as follows:

1. draw Discrete distributions σ_{zw} from a Dirichlet prior δ for each topic z and each word w;

2. for each document d, draw a Discrete distribution $\theta^{(d)}$ from a Dirichlet prior α; then for each word $w_i^{(d)}$ in document d:

   (a) draw $z_i^{(d)}$ from Discrete $\theta^{(d)}$; and

   (b) draw $w_i^{(d)}$ from Discrete $\sigma_{z_i^{(d)} w_{i-1}^{(d)}}$.

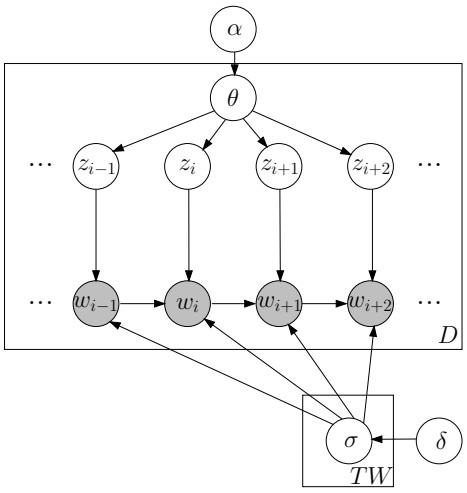
SYMBOL	DESCRIPTION
T	number of topics
D	number of documents
W	number of unique words
N_d	number of word tokens in document d
$z_i^{(d)}$	the topic associated with the i^{th} token in the document d
$x_i^{(d)}$	the bigram status between the $(i-1)^{th}$ token and i^{th} token in the document d
$w_i^{(d)}$	the i^{th} token in document d
$\theta^{(d)}$	the multinomial (Discrete) distribution of topics w.r.t. the document d
ϕ_z	the multinomial (Discrete) unigram distribution of words w.r.t. topic z
ψ_v	in Figure 1(b), the binomial (Bernoulli) distribution of status variables w.r.t. previous word v
ψ_{zv}	in Figure 1(c), the binomial (Bernoulli) distribution of status variables w.r.t. previous topic z/word v
σ_{zv}	in Figure 1(a) and (c), the multinomial (Discrete) bigram distribution of words w.r.t. topic z/word v
σ_v	in Figure 1(b), the multinomial (Discrete) bigram distribution of words w.r.t. previous word v
α	Dirichlet prior of θ
β	Dirichlet prior of ϕ
γ	Dirichlet prior of ψ
δ	Dirichlet prior of σ

Table 1. Notation used in this paper

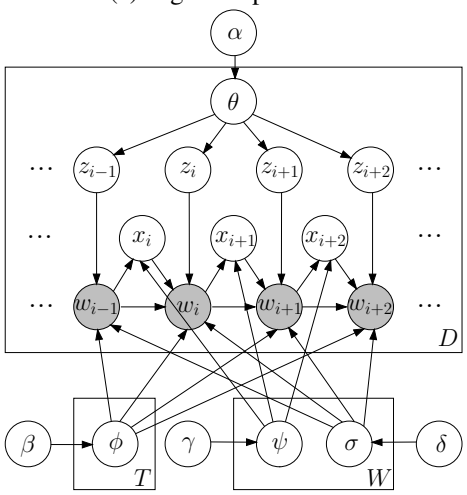
2.2 LDA Collocation Model (LDACOL)

Starting from the LDA topic model, the LDA collocation model [20] (not yet published) introduces a new set of random variables (for bigram status) $\mathbf{x}$ ($x_i = 1$: w_{i-1} and w_i form a bigram; $x_i = 0$: they do not) that denote if a bigram can be formed with the previous token, in addition to the two sets of random variables $\mathbf{z}$ and $\mathbf{w}$ in LDA. Thus, it has the power to decide if to generate a bigram or a unigram. At this aspect, it is more realistic than the bigram topic model which always generates bigrams. After all, unigrams are the major components in a document. We assume the status variable x_1 is observed, and only a unigram is allowed at the beginning of a document. If we want to put more constraints into the model (e.g., no bigram is allowed for sentence/paragraph boundary; only a unigram can be considered for the next word after a stop word is removed; etc.), we can assume that the corresponding status variables are observed as well. This model's graphical model presentation is shown in Figure 1(b).

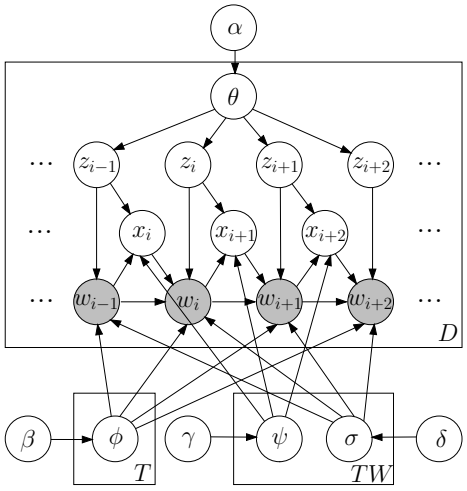
The generative process of the LDA collocation model is described as follows:



(a) Bigram topic model



(b) LDA-Collocation model



(c) Topical n-gram model

Figure 1. Three n-gram based topic models

1. draw Discrete distributions ϕ_z from a Dirichlet prior β for each topic z;

2. draw Bernoulli distributions ψ_w from a Beta prior γ for each word w;

3. draw Discrete distributions σ_w from a Dirichlet prior δ for each word w;

4. for each document d, draw a Discrete distribution $\theta^{(d)}$ from a Dirichlet prior α; then for each word $w_i^{(d)}$ in document d:

   (a) draw $x_i^{(d)}$ from Bernoulli $\psi_{w_{i-1}^{(d)}}$;

   (b) draw $z_i^{(d)}$ from Discrete $\theta^{(d)}$; and

   (c) draw $w_i^{(d)}$ from Discrete $\sigma_{w_{i-1}^{(d)}}$ if $x_i^{(d)} = 1$; else draw $w_i^{(d)}$ from Discrete $\phi_{z_i^{(d)}}$.

Note that in the LDA Collocation model, bigrams do not have topics as the second term of a bigram is generated from a distribution σ_v conditioned on the previous word v only.

2.3 Topical N-gram Model (TNG)

The topical n-gram model (TNG) is not a pure addition of the bigram topic model and LDA collocation model. It can solve the problem associated with the "neural network" example as the bigram topic model, and automatically determine whether a composition of two terms is indeed a bigram as in the LDA collocation model. However, like other collocation discovery methods discussed in Section 3, a discovered bigram is always a bigram in the LDA Collocation model no matter what the context is.

One of the key contributions of our model is to make it possible to decide whether to form a bigram for the same two consecutive word tokens depending on their nearby context (i.e., co-occurrences). Thus, additionally, our model is a perfect solution for the "white house" example in Section 1. As in the LDA collocation model, we may assume some $\mathbf{x}$'s are observed for the same reason as we discussed in Section 2.2. The graphical model presentation of this model is shown in Figure 1(c). Its generative process can be described as follows:

1. draw Discrete distributions ϕ_z from a Dirichlet prior β for each topic z;

2. draw Bernoulli distributions ψ_{zw} from a Beta prior γ for each topic z and each word w;

3. draw Discrete distributions σ_{zw} from a Dirichlet prior δ for each topic z and each word w;

4. for each document d, draw a Discrete distribution $\theta^{(d)}$ from a Dirichlet prior α; then for each word $w_i^{(d)}$ in document d:

   (a) draw $x_i^{(d)}$ from Bernoulli $\psi_{z_{i-1}^{(d)} w_{i-1}^{(d)}}$;

   (b) draw $z_i^{(d)}$ from Discrete $\theta^{(d)}$; and

   (c) draw $w_i^{(d)}$ from Discrete $\sigma_{z_i^{(d)} w_{i-1}^{(d)}}$ if $x_i^{(d)} = 1$; else draw $w_i^{(d)}$ from Discrete $\phi_{z_i^{(d)}}$.

Note that our model is a more powerful generalization of BTM and of LDACOL. Both BTM (by setting all x's to 1) and LDACOL (by making σ conditioned on previous word only) are the special cases of our TNG models.

Before discussing the inference problem of our model, let us pause for a brief interlude on topic consistency of terms in a bigram. As shown in the above, the topic assignments for the two terms in a bigram are not required to be identical. We can take the topic of the first/last word token or the most common topic in the phrase, as the topic of the phrase. In this paper, we will use the topic of the last term as the topic of the phrase for simplicity, since long noun phrases do truly sometimes have components indicative of different topics, and its last noun is usually the "head noun". Alternatively, we could enforce consistency in the model with ease, by simply adding two more sets of arrows ($z_{i-1} \rightarrow z_i$ and $x_i \rightarrow z_i$). Accordingly, we could substitute Step 4(b) in the above generative process with "draw $z_i^{(d)}$ from Discrete $\theta^{(d)}$ if $x_i^{(d)} = 1$; else let $z_i^{(d)} = z_{i-1}^{(d)}$;" In this way, a word has the option to inherit a topic assignment from its previous word if they form a bigram phrase. However, from our experimental results, the first choice yields better performance. From now on, we will focus on the model shown in Figure 1(c).

Finally we want to point out that the topical n-gram model is not only a new framework for distilling n-gram phrases depending on nearby context, but also a more sensible topic model than the ones using word co-occurrences alone.

In state-of-the-art hierarchical Bayesian models such as latent Dirichlet allocation, exact inference over hidden topic variables is typically intractable due to the large number of latent variables and parameters in the models. Approximate inference techniques such as variational methods [12], Gibbs sampling [1] and expectation propagation [17] have been developed to address this issue. We use Gibbs sampling to conduct approximate inference in this paper. To reduce the uncertainty introduced by θ, ϕ, ψ, and σ, we could integrate them out with no trouble because of the conjugate prior setting in our model. Starting from the joint distribution $P(\mathbf{w}, \mathbf{z}, \mathbf{x}|\alpha, \beta, \gamma, \delta)$, we can work out the conditional probabilities $P(z_i^{(d)}, x_i^{(d)}|\mathbf{z}_{-i}^{(d)}, \mathbf{x}_{-i}^{(d)}, \mathbf{w}, \alpha, \beta, \gamma, \delta)$ conveniently[1] using Bayes rule, where $\mathbf{z}_{-i}^{(d)}$ denotes the topic assignments for all word tokens except word $w_i^{(d)}$, and $\mathbf{x}_{-i}^{(d)}$ represents the bigram status for all tokens except word $w_i^{(d)}$. During Gibbs sampling, we draw the topic assignment $z_i^{(d)}$ and the bigram status $x_i^{(d)}$ iteratively[2] for each word token $w_i^{(d)}$ according to the following conditional probability distribution:

$$
\begin{aligned}
& P(z_i^{(d)}, x_i^{(d)}|\mathbf{z}_{-i}^{(d)}, \mathbf{x}_{-i}^{(d)}, \mathbf{w}, \alpha, \beta, \gamma, \delta) \\
\propto\ & (\gamma_{x_i^{(d)}} + p_{z_{i-1}^{(d)} w_{i-1}^{(d)} x_i} - 1)(\alpha_{z_i^{(d)}} + q_{dz_i^{(d)}} - 1) \\
& \times \begin{cases} \frac{\beta_{w_i^{(d)}} + n_{z_i^{(d)} w_i^{(d)}} - 1}{\sum_{v=1}^{W} (\beta_v + n_{z_i^{(d)} v}) - 1} & \text{if } x_i^{(d)} = 0 \\ \frac{\delta_{w_i^{(d)}} + m_{z_i^{(d)} w_{i-1}^{(d)} w_i^{(d)}} - 1}{\sum_{v=1}^{W} (\delta_v + m_{z_i^{(d)} w_{i-1}^{(d)} v}) - 1} & \text{if } x_i^{(d)} = 1 \end{cases}
\end{aligned}
$$

where n_{zw} represents how many times word w is assigned into topic z as a unigram, m_{zwv} represents how many times word v is assigned to topic z as the 2^{nd} term of a bigram given the previous word w, p_{zwk} denotes how many times the status variable $x = k$ (0 or 1) given the previous word w and the previous word's topic z, and q_{dz} represents how many times a word is assigned to topic z in document d. Note all counts here do include the assignment of the token being visited. Details of the Gibbs sampling derivation are provided in Appendix A.

Simple manipulations give us the posterior estimates of θ, ϕ, ψ, and σ as follows:

$$
\begin{aligned}
\hat{\theta}_z^{(d)} &= \frac{\alpha_z + q_{dz}}{\sum_{t=1}^{T} (\alpha_t + q_{dt})} & \hat{\phi}_{zw} &= \frac{\beta_w + n_{zw}}{\sum_{v=1}^{W} (\beta_v + n_{zv})} \\
\hat{\psi}_{zwk} &= \frac{\gamma_k + p_{zwk}}{\sum_{k=0}^{1} (\gamma_k + p_{zwk})} & \hat{\sigma}_{zwv} &= \frac{\delta_v + m_{zwv}}{\sum_{v=1}^{W} (\delta_v + m_{zwv})}
\end{aligned} \tag{1}
$$

As discussed in the bigram topic model [22], one could certainly infer the values of the hyperparameters in TNG using a Gibbs EM algorithm [1]. For many applications, topic models are sensitive to hyperparameters, and it is important to get the right values for the hyperparameters. In the particular experiments discussed in this paper, however, we find that sensitivity to hyperparameters is not a big concern. For simplicity and feasibility in our Gigabyte TREC retrieval tasks, we skip the inference of hyperparameters, and use some reported empirical values for them instead to show salient results.

3 Related Work

Collocation has long been studied by lexicographers and linguists in various ways. Traditional collocation discov-

[1] As shown in Appendix A, one could further calculate $P(z_i^{(d)}|\cdots)$ and $P(x_i^{(d)}|\cdots)$ as in a traditional Gibbs sampling procedure.

[2] For some observed $x_i^{(d)}$, only $z_i^{(d)}$ needs to be drawn.

ery methods range from frequency to variance, to hypothesis testing, to mutual information. The simplest method is counting. A small amount of linguistic knowledge (a part-of-speech filter) has been combined with frequency [13] to discover surprisingly meaningful phrases. Variance based collocation discovery [19] considers collocations in a more flexible way than fixed phrases. However, high frequency and low variance can be accidental. Hypothesis testing can be used to assess whether or not two words occur together more often than chance. Many statistical tests have been explored, for example, t-test [5], χ^2 test [4], and likelihood ratio test [7]. More recently, an information-theoretically motivated method for collocation discovery is utilizing mutual information [6, 11].

The hierarchical Dirichlet language model [14] is closely related to the bigram topic model [22]. The probabilistic view of smoothing in language models shows how to take advantage of a bigram model in a Bayesian way.

The main stream of topic modeling has gradually gained a probabilistic flavor as well in the past decade. One of the most popular topic model, latent Dirichlet allocation (LDA), which makes the bag-of-words assumption, has made a big impact in the fields of natural language processing, statistical machine learning and text mining. Three models we discussed in Section 2 all contain an LDA component that is responsible for the topic part.

In our point of view, the HMMLDA model [10] is the first attack to word dependency in the topic modeling framework. The authors present HMMLDA as a generative composite model that takes care of both short-range syntactic dependencies and long-range semantic dependencies between words; its syntactic part is a hidden Markov model and the semantic component is a topic model (LDA). Interesting results based on this model are shown on tasks such as part-of-speech tagging and document classification.

4 Experimental Results

We apply the topical n-gram model to the NIPS proceedings dataset that consists of the full text of the 13 years of proceedings from 1987 to 1999 Neural Information Processing Systems (NIPS) Conferences. In addition to downcasing and removing stopwords and numbers, we also removed the words appearing less than five times in the corpus—many of them produced by OCR errors. Two-letter words (primarily coming from equations), were removed, except for "ML", "AI", "KL", "BP", "EM" and "IR." The dataset contains 1,740 research papers, 13,649 unique words, and 2,301,375 word tokens in total. Topics found from a 50-topic run on the NIPS dataset (10,000 Gibbs sampling iterations, with symmetric priors $\alpha = 1$, $\beta = 0.01$, $\gamma = 0.1$, and $\delta = 0.01$) of the topical n-gram model are shown in Table 2 as anecdotal evidence, with comparison to the corresponding closest (by KL divergence) topics found by LDA.

The "Reinforcement Learning" topic provides an extremely salient summary of the corresponding research area. The LDA topic assembles many common words used in reinforcement learning, but in its word list, there are quite a few generic words (such as "function", "dynamic", "decision") that are common and highly probable in many other topics as well. In TNG, we can find that these generic words are associated with other words to form n-gram phrases (such as "markov decision process", etc.) that are only highly probable in reinforcement learning. More importantly, by forming n-gram phrases, the unigram word list produced by TNG is also cleaner. For example, because of the prevalence of generic words in LDA, highly related words (such as "q-learning" and "goal") are not ranked high enough to be shown in the top 20 word list. On the contrary, they are ranked very high in the TNG's unigram word list.

In the other three topics (Table 2), we can find similar phenomena as well. For example, in "Human Receptive System", some generic words (such as "field", "receptive") are actually the components of the popular phrases in this area as shown in the TNG model. "system" is ranked high in LDA, but almost meaningless, and on the other hand, it does not appear in the top word lists of TNG. Some extremely related words (such as "spatial"), ranked very high in TNG, are absent in LDA's top word list. In "Speech Recognition", the dominating generic words (such as "context", "based", "set", "probabilities", "database") make the LDA topic less understandable than even just TNG's unigram word list.

In many situations, a crucially related word might be not mentioned enough to be clearly captured in LDA, on the other hand, it would become very salient as a phrase due to the relatively stronger co-occurrence pattern in an extremely sparse setting for phrases. The "Support Vector Machines" topic provides such an example. We can imagine that "kkt" will be mentioned no more than a few times in a typical NIPS paper, and it probably appears only as a part of the phrase "kkt conditions". TNG satisfyingly captures it successfully as a highly probable phrase in the SVM topic.

As we discussed before, higher-order n-grams ($n > 2$) can be approximately modeled by concatenating consecutive bigrams in the TNG model, as shown in Table 2 (such as "markov decision process", "hidden markov model" and "support vector machines", etc.).

To numerically evaluate the topical n-gram model, we could have used some standard measures such as perplexity and document classfication accuracy. However, to convincingly illustrate the power of the TNG model on larger, more real scale, here we apply the TNG model to a much larger standard text mining task—we employ the TNG model within the language modeling framework to conduct ad-hoc retrieval on Gigabyte TREC collections.

Reinforcement Learning			**Human Receptive System**		
LDA	n-gram (2+)	n-gram (1)	LDA	n-gram (2+)	n-gram (1)
state	reinforcement learning	action	motion	receptive field	motion
learning	optimal policy	policy	visual	spatial frequency	spatial
policy	dynamic programming	reinforcement	field	temporal frequency	visual
action	optimal control	states	position	visual motion	receptive
reinforcement	function approximator	actions	figure	motion energy	response
states	prioritized sweeping	function	direction	tuning curves	direction
time	finite-state controller	optimal	fields	horizontal cells	cells
optimal	learning system	learning	eye	motion detection	figure
actions	reinforcement learning rl	reward	location	preferred direction	stimulus
function	function approximators	control	retina	visual processing	velocity
algorithm	markov decision problems	agent	receptive	area mt	contrast
reward	markov decision processes	q-learning	velocity	visual cortex	tuning
step	local search	goal	vision	light intensity	moving
dynamic	state-action pair	space	moving	directional selectivity	model
control	markov decision process	step	system	high contrast	temporal
sutton	belief states	environment	flow	motion detectors	responses
rl	stochastic policy	system	edge	spatial phase	orientation
decision	action selection	problem	center	moving stimuli	light
algorithms	upright position	steps	light	decision strategy	stimuli
agent	reinforcement learning methods	transition	local	visual stimuli	cell

Speech Recognition			**Support Vector Machines**		
LDA	n-gram (2+)	n-gram (1)	LDA	n-gram (2+)	n-gram (1)
recognition	speech recognition	speech	kernel	support vectors	kernel
system	training data	word	linear	test error	training
word	neural network	training	vector	support vector machines	support
face	error rates	system	support	training error	margin
context	neural net	recognition	set	feature space	svm
character	hidden markov model	hmm	nonlinear	training examples	solution
hmm	feature vectors	speaker	data	decision function	kernels
based	continuous speech	performance	algorithm	cost functions	regularization
frame	training procedure	phoneme	space	test inputs	adaboost
segmentation	continuous speech recognition	acoustic	pca	kkt conditions	test
training	gamma filter	words	function	leave-one-out procedure	data
characters	hidden control	context	problem	soft margin	generalization
set	speech production	systems	margin	bayesian transduction	examples
probabilities	neural nets	frame	vectors	training patterns	cost
features	input representation	trained	solution	training points	convex
faces	output layers	sequence	training	maximum margin	algorithm
words	training algorithm	phonetic	svm	strictly convex	working
frames	test set	speakers	kernels	regularization operators	feature
database	speech frames	mlp	matrix	base classifiers	sv
mlp	speaker dependent	hybrid	machines	convex optimization	functions

Table 2. The four topics from a 50-topic run of TNG on 13 years of NIPS research papers with their closest counterparts from LDA. The Title above the word lists of each topic is our own summary of the topic. To better illustrate the difference between TNG and LDA, we list the n-grams ($n > 1$) and unigrams separately for TNG. Each topic is shown with the 20 sorted highest-probability words. The TNG model produces clearer word list for each topic by associating many generic words (such as "set", "field", "function", etc.) with other words to form n-gram phrases.

4.1 Ad-hoc Retrieval

Traditional information retrieval (IR) models usually represent text with bags-of-words assuming that words occur independently, which is not exactly appropriate to natural language. To address this problem, researchers have been working on capturing word dependencies. There are mainly two types of dependencies being studied and shown to be effective: 1) topical (semantic) dependency, which is also called long-distance dependency. Two words are considered dependent when their meanings are related and they co-occur often, such as "fruit" and "apple". Among models capturing semantic dependency, the LDA-based document models [23] are state-of-the-art. For IR applications, a major advantage of topic models (document expansion), compared to online query expansion in pseudo relevance feedback, is that they can be trained offline, thus more efficient in handling a new query; 2) phrase dependency, also called short-distance dependency. As reported in literature, retrieval performance can be boosted if the similarity between a user query and a document is calculated by common phrases instead of common words [9, 8, 21, 18]. Most research on phrases in information retrieval has employed an independent collocation discovery module, e.g., using the methods described in Section 3. In this way, a phrase can be indexed exactly as an ordinary word.

The topical n-gram model automatically and simultaneously takes cares of both semantic co-occurrences and phrases. Also, it does not need a separate module for phrase discovery, and everything can be seamlessly integrated into the language modeling framework, which is one of the most popular statistically principled approaches to IR. In this section, we illustrate the difference in IR experiments of the TNG and LDA models, and compare the IR performance of all three models in Figure 1 on a TREC collection introduced below.

The SJMN dataset, taken from TREC with standard queries 51-150 that are taken from the *title* field of TREC topics, covers materials from San Jose Mercury News in 1991. All text is downcased and only alphabetic characters are kept. Stop words in both the queries and documents are removed, according to a common stop word list in the Bow toolkit [16]. If any two consecutive tokens were originally separated by a stopword, no bigram is allowed to be formed. In total, the SJMN dataset we use contains 90,257 documents, 150,714 unique words, and 21,156,378 tokens, which is order of magnitude larger than the NIPS dataset. Relevance judgments are taken from the the judged pool of the top retrieved documents by various participating retrieval systems from previous TREC conferences.

The number of topics are set to be 100 for all models with 10,000 Gibbs sampling iterations, and the same hyperparameter setting (with symmetric priors $\alpha = 1$, $\beta = 0.01$, $\gamma = 0.1$, and $\delta = 0.01$) for the NIPS dataset are used. Here, we aim to beat the state-of-the-art model [23] instead of the state-of-the-art results in TREC retrieval that need significant, non-modeling effort to achieve (such as stemming).

4.2 Difference between Topical N-grams and LDA in IR Applications

From both of LDA and TNG, a word distribution for each document can be calculated, which thus can be viewed as a document model. With these distributions, the likelihood of generating a query can be computed to rank documents, which is the basic idea in the query likelihood (QL) model in IR. When the two models are directly applied to do ad-hoc retrieval, the TNG model performs significant better than the LDA model under the Wilcoxon test at 95% level. Among of 4881 relevant documents for all queries, LDA retrieves 2257 of them but TNG gets 2450, 8.55% more. The average precision for TNG is 0.0709, 61.96% higher than its LDA counterpart (0.0438). Although these results are not the state-of-the-art IR performance, we claim that, if used alone, TNG represent a document better than LDA. The average precisions for both models are very low, because corpus-level topics may be too coarse to be used as the only representation in IR [3, 23]. Significant improvements in IR can be achieved through a combination with the basic query likelihood model.

In the query likelihood model, each document is scored by the likelihood of its model generating a query Q, $P_{LM}(Q|d)$. Let the query $Q = (q_1, q_2, ..., q_{L_Q})$. Under the bag-of-words assumption, $P_{LM}(Q|d) = \prod_{i=1}^{L_Q} P(q_i|d)$, which is often specified by the document model with Dirichlet smoothing [24],

$$P_{LM}(q|d) = \frac{N_d}{N_d + \mu} P_{ML}(q|d) + (1 - \frac{N_d}{N_d + \mu}) P_{ML}(q|\text{coll}),$$

where N_d is the length of document d, $P_{ML}(q|d)$ and $P_{ML}(q|\text{coll})$ are the maximum likelihood (ML) estimates of a query term q generated in document d and in the entire collection, respectively, and μ is the Dirichlet smoothing prior (in our reported experiments we used a fixed value with $\mu = 1000$ as in [23]).

To calculate the query likelihood from the TNG model within the language modeling framework, we need to sum over the topic variable and bigram status variable for each token in the query token sequence. Given the posterior estimates $\hat{\theta}$, $\hat{\phi}$, $\hat{\psi}$, and $\hat{\sigma}$ (Equation 1), the query likelihood of query Q given document d, $P_{TNG}(Q|d)$ can be calculated[3] as

$$P_{TNG}(Q|d) = \prod_{i=1}^{L_Q} P_{TNG}(q_i|q_{i-1}, d),$$

[3] A dummy q_0 is assumed at the beginning of every query, for the convenience of mathematical presentation.

No.	Query	LDA	TNG	Change
053	Leveraged Buyouts	0.2141	0.3665	71.20%
097	Fiber Optics Applications	0.1376	0.2321	68.64%
108	Japanese Protectionist Measures	0.1163	0.1686	44.94%
111	Nuclear Proliferation	0.2353	0.4952	110.48%
064	Hostage-Taking	0.4265	0.4458	4.52%
125	Anti-smoking Actions by Government	0.3118	0.4535	45.47%
145	Influence of the "Pro-Israel Lobby"	0.2900	0.2753	-5.07%
148	Conflict in the Horn of Africa	0.1990	0.2788	40.12%

Table 3. Comparison of LDA and TNG on TREC retrieval performance (average precision) of eight queries. The top four queries obviously contain phrase(s), and thus TNG achieves much better performance. On the other hand, the bottom four queries do not contain common phrase(s) after preprocessing (stopping and punctuation removal). Surprisingly, TNG still outperforms LDA on some of these queries.

where

$$P_{TNG}(q_i|q_{i-1},d) = \sum_{z_i=1}^{T}(P(x_i=0|\hat{\psi}_{q_{i-1}})P(q_i|\hat{\phi}_{z_i}) + P(x_i=1|\hat{\psi}_{q_{i-1}})P(q_i|\hat{\sigma}_{z_i q_{i-1}}))P(z_i|\hat{\theta}^{(d)}),$$

and,

$$P(x_i|\hat{\psi}_{q_{i-1}}) = \sum_{z_{i-1}=1}^{T} P(x_i|\hat{\psi}_{z_{i-1}q_{i-1}})P(z_{i-1}|\hat{\theta}^{(d)}).$$

Due to stopping and punctuation removal, we may simply set $P(x_i=0|\hat{\psi}_{q_{i-1}})=1$ and $P(x_i=1|\hat{\psi}_{q_{i-1}})=0$ at corresponding positions in a query. Note here in the above calculation, the bag-of-words assumption is not made any more.

Similar to the method in [23], we can combine the query likelihood from the basic language model and the likelihood from the TNG model in various ways. One can combine them at query level, i.e.,

$$P(Q|d) = \lambda P_{LM}(Q|d) + (1-\lambda)P_{TNG}(Q|d),$$

where λ is a weighting factor between the two likelihoods.

Alternatively, under first order Markov assumption, $P(Q|d) = P(q_1|d)\prod_{i=2}^{L_Q} P(q_i|q_{i-1},d)$, and one can combine the query likelihood at query term level (used in this paper), that is,

$$P(q_i|q_{i-1},d) = \lambda P_{LM}(q_i|d) + (1-\lambda)P_{TNG}(q_i|q_{i-1},d).$$

To illustrate the difference of TNG and LDA in IR applications, we select a few of the 100 queries that clearly contain phrase(s), and another few of them that do not contain phrase due to stopping and puntuation removal, on which we compare the IR performance (average precision)[4] as shown in Table 3.

[4]The results reported in [23] is a little better since they did stemming.

4.3 Comparison of BTM, LDACOL and TNG on TREC Ad-hoc Retrieval

In this section, we compare the IR performance of the three n-gram based topic models on the SJMN dataset[5], as shown in Table 4. For a fair comparison, the weighting factor λ (reported in Table 4) are independently chosen to get the best performance from each model. Under the Wilcoxon test with 95% confidence, TNG significantly outperforms BTM and LDACOL on this standard retrieval task.

Space limitations prevent us from presenting the results for all queries, but it is interesting to see that different models are good at quite different queries. For some queries (such as No. 117 and No. 138), TNG and BTM perform similarly, and better than LDACOL, and for some other queries (such as No. 110 and No. 150), TNG and LDACOL perform similarly, and better than BTM. There are also queries (such as No. 061 and No. 130) for which TNG performs better than both BTM and LDACOL. We believe that they are clear empirical evidence that our TNG model are more generic and powerful than BTM and LDACOL.

We analyze the performance of the TNG model for query No. 061, as an example. As we inspect the phrase "Iran-Contra" contained in the query, we find that it has been primarily assigned to two topics (politics and economy) in TNG. This has increased the bigram likelihood of some documents emphasizing the relevant topic (such as "SJMN91-06263203"), thus helps promote these documents to higher ranks. As a special case of TNG, LDACOL is unable to capture this and leads to inferior performance.

It is true that for certain queries (such as No. 069 and No. 146), TNG performs worse than BTM and LDACOL, but we notice that all models perform badly on these queries

[5]The running times of our C implementation on a dual-processor Opteron for the three models are 11.5, 17, 22.5 hours, respectively.

No.	Query	TNG	BTM	Change	LDACOL	Change
061	Israeli Role in Iran-Contra Affair	0.1635	0.1104	-32.47%	0.1316	-19.49%
069	Attempts to Revive the SALT II Treaty	0.0026	0.0071	172.34%	0.0058	124.56%
110	Black Resistance Against the South African Government	0.4940	0.3948	-20.08%	0.4883	-1.16%
117	Capacity of the U.S. Cellular Telephone Network	0.2801	0.3059	9.21%	0.1999	-28.65%
130	Jewish Emigration and U.S.-USSR Relations	0.2087	0.1746	-16.33%	0.1765	-15.45%
138	Iranian Support for Lebanese Hostage-takers	0.4398	0.4429	0.69%	0.3528	-19.80%
146	Negotiating an End to the Nicaraguan Civil War	0.0346	0.0682	97.41%	0.0866	150.43%
150	U.S. Political Campaign Financing	0.2672	0.2323	-13.08%	0.2688	0.59%
	All Queries	0.2122	0.1996	-5.94%*	0.2107	-0.73%*

Table 4. Comparison of the bigram topic model ($\lambda = 0.7$), LDA collocation model ($\lambda = 0.9$) and the topical n-gram Model ($\lambda = 0.8$) on TREC retrieval performance (average precision). * indicates statistically significant differences in performance with 95% confidence according to the Wilcoxon test. TNG performs significantly better than other two models overall.

and the behaviors are more possibly due to randomness.

5 Conclusions

In this paper, we have presented the topical n-gram model. The TNG model automatically determines to form an n-gram (and further assign a topic) or not, based on its surrounding context. Examples of topics found by TNG are more interpretable than its LDA counterpart. We also demonstrate how TNG can help improve retrieval performance in standard ad-hoc retrieval tasks on TREC collections over its two special-case n-gram based topic models.

Unlike some traditional phrase discovery methods, the TNG model provides a systematic way to model (topical) phrases and can be seamlessly integrated with many probabilistic frameworks for various tasks such as phrase discovery, ad-hoc retrieval, machine translation, speech recognition and statistical parsing.

Evaluating n-gram based topic models is a big challenge. As reported in [22], the bigram topic models have only been shown to be effective on hundreds of documents, and also we have not seen a formal evaluation of the unpublished LDA collocation models. To the best of our knowledge, our paper presents the very first application of all three n-gram based topic models on Gigabyte collections, and a novel way to integrate n-gram based topic models into the language modeling framework for information retrieval tasks.

Appendix

A Gibbs Sampling Derivation for the Topical N-grams Model

We begin with the joint distribution $P(\mathbf{w}, \mathbf{x}, \mathbf{z}|\alpha, \beta, \gamma, \delta)$. We can take advantage of conjugate priors to simplify the integrals. All symbols are defined in Section 2.

$$
\begin{aligned}
& P(\mathbf{w}, \mathbf{z}, \mathbf{x}|\alpha, \beta, \gamma, \delta) \\
= & \iiint \prod_{d=1}^{D} \prod_{i=1}^{N_d} (P(w_i^{(d)}|x_i^{(d)}, \phi_{z_i^{(d)}}, \sigma_{z_i^{(d)} w_{i-1}^{(d)}}) \\
& P(x_i^{(d)}|\psi_{z_{i-1}^{(d)} w_{i-1}^{(d)}})) \prod_{z=1}^{T} \prod_{v=1}^{W} p(\sigma_{zv}|\delta) p(\psi_{zv}|\gamma) \mathrm{d}\Sigma \mathrm{d}\Psi \\
& \prod_{z=1}^{T} p(\phi_z|\beta) \mathrm{d}\Phi \int \prod_{d=1}^{D} \left(\prod_{i=1}^{N_d} P(z_i^{(d)}|\theta_d) p(\theta_d|\alpha) \right) \mathrm{d}\Theta \\
= & \int \prod_{z=1}^{T} \left(\prod_{v=1}^{W} \phi_{zv}^{n_{zv}} \frac{\Gamma(\sum_{v=1}^{W} \beta_v)}{\prod_{v=1}^{W} \Gamma(\beta_v)} \prod_{v=1}^{W} \phi_{zv}^{\beta_v - 1} \right) \mathrm{d}\Phi \\
& \times \int \prod_{z=1}^{T} \prod_{w=1}^{W} \left(\prod_{v=1}^{W} \sigma_{zwv}^{m_{zwv}} \frac{\Gamma(\sum_{v=1}^{W} \delta_v)}{\prod_{v=1}^{W} \Gamma(\delta_v)} \prod_{v=1}^{W} \sigma_{zwv}^{\delta_v - 1} \right) \mathrm{d}\Sigma \\
& \times \int \prod_{z=1}^{T} \prod_{w=1}^{W} \left(\prod_{k=0}^{1} \psi_{zwk}^{p_{zwk}} \frac{\Gamma(\sum_{k=0}^{1} \gamma_v)}{\prod_{k=0}^{1} \Gamma(\gamma_k)} \prod_{k=0}^{1} \psi_{zwk}^{\gamma_k - 1} \right) \mathrm{d}\Psi \\
& \times \int \prod_{d=1}^{D} \left(\prod_{z=1}^{T} \theta_{dz}^{q_{dz}} \frac{\Gamma(\sum_{z=1}^{T} \alpha_z)}{\prod_{z=1}^{T} \Gamma(\alpha_z)} \prod_{z=1}^{T} \theta_{dz}^{\alpha_z - 1} \right) \mathrm{d}\Theta \\
\propto & \prod_{z=1}^{T} \frac{\prod_{v=1}^{W} \Gamma(n_{zv} + \beta_v)}{\Gamma(\sum_{v=1}^{W} (n_{zv} + \beta_v))} \prod_{z=1}^{T} \prod_{w=1}^{W} \frac{\prod_{v=1}^{W} \Gamma(m_{zwv} + \delta_v)}{\Gamma(\sum_{v=1}^{W} (m_{zwv} + \delta_v))} \\
& \prod_{z=1}^{T} \prod_{w=1}^{W} \frac{\prod_{k=0}^{1} \Gamma(p_{zwk} + \gamma_k)}{\Gamma(\sum_{k=0}^{1} (p_{zwk} + \gamma_k))} \prod_{d=1}^{D} \frac{\prod_{z=1}^{T} \Gamma(q_{dz} + \alpha_z)}{\Gamma(\sum_{z=1}^{T} (q_{dz} + \alpha_z))}
\end{aligned}
$$

Using the chain rule and $\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1)$, we can obtain the conditional probability conveniently,

$$P(z_i^{(d)}, x_i^{(d)}|\mathbf{w}, \mathbf{z}_{-i}^{(d)}, \mathbf{x}_{-i}^{(d)}, \alpha, \beta, \gamma, \delta)$$

$$
\begin{aligned}
&= \frac{P(w_i^{(d)}, z_i^{(d)}, x_i^{(d)} | \mathbf{w}_{-i}^{(d)}, \mathbf{z}_{-i}^{(d)}, \mathbf{x}_{-i}^{(d)}, \alpha, \beta, \gamma, \delta)}{P(w_i^{(d)} | \mathbf{w}_{-i}^{(d)}, \mathbf{z}_{-i}^{(d)}, \mathbf{x}_{-i}^{(d)}, \alpha, \beta, \gamma, \delta)} \\
&\propto (\gamma_{x_i^{(d)}} + p_{z_{i-1}^{(d)} w_{i-1}^{(d)} x_i} - 1)(\alpha_{z_i^{(d)}} + q_{d z_i^{(d)}} - 1) \\
&\quad \times \begin{cases} \frac{\beta_{w_i^{(d)}} + n_{z_i^{(d)} w_i^{(d)}} - 1}{\sum_{v=1}^{W} (\beta_v + n_{z_i^{(d)} v}) - 1} & \text{if } x_i^{(d)} = 0 \\ \frac{\delta_{w_i^{(d)}} + m_{z_i^{(d)} w_{i-1}^{(d)} w_i^{(d)}} - 1}{\sum_{v=1}^{W} (\delta_v + m_{z_i^{(d)} w_{i-1}^{(d)} v}) - 1} & \text{if } x_i^{(d)} = 1 \end{cases}
\end{aligned}
$$

Or equivalently,

$$
\begin{aligned}
&P(z_i^{(d)} | \mathbf{w}, \mathbf{z}_{-i}^{(d)}, \mathbf{x}, \alpha, \beta, \gamma, \delta) \\
&\propto (\alpha_{z_i^{(d)}} + q_{d z_i^{(d)}} - 1) \\
&\quad \times \begin{cases} \frac{\beta_{w_i^{(d)}} + n_{z_i^{(d)} w_i^{(d)}} - 1}{\sum_{v=1}^{W} (\beta_v + n_{z_i^{(d)} v}) - 1} & \text{if } x_i^{(d)} = 0 \\ \frac{\delta_{w_i^{(d)}} + m_{z_i^{(d)} w_{i-1}^{(d)} w_i^{(d)}} - 1}{\sum_{v=1}^{W} (\delta_v + m_{z_i^{(d)} w_{i-1}^{(d)} v}) - 1} & \text{if } x_i^{(d)} = 1 \end{cases}
\end{aligned}
$$

And,

$$
\begin{aligned}
&P(x_i^{(d)} | \mathbf{w}, \mathbf{z}, \mathbf{x}_{-i}^{(d)}, \alpha, \beta, \gamma, \delta) \\
&\propto (\gamma_{x_i^{(d)}} + p_{z_{i-1}^{(d)} w_{i-1}^{(d)} x_i} - 1) \\
&\quad \times \begin{cases} \frac{\beta_{w_i^{(d)}} + n_{z_i^{(d)} w_i^{(d)}} - 1}{\sum_{v=1}^{W} (\beta_v + n_{z_i^{(d)} v}) - 1} & \text{if } x_i^{(d)} = 0 \\ \frac{\delta_{w_i^{(d)}} + m_{z_i^{(d)} w_{i-1}^{(d)} w_i^{(d)}} - 1}{\sum_{v=1}^{W} (\delta_v + m_{z_i^{(d)} w_{i-1}^{(d)} v}) - 1} & \text{if } x_i^{(d)} = 1 \end{cases}
\end{aligned}
$$

References

[1] C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003.

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

[3] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *Advances in Neural Information Processing Systems 19*, pages 241–248, 2007.

[4] K. Church and W. Gale. Concordances for parallel text. In *Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research*, pages 40–62, 1991.

[5] K. Church and P. Hanks. Word association norms, mutual information and lexicography. In *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 76–83, 1989.

[6] K. W. Church, W. Gale, P. Hanks, and D. Hindle. Using statistics in lexical analysis. In *Lexical Acquisition: Using On-line Resources to Build a Lexicon*, pages 115–164. Lawrence Erlbaum, 1991.

[7] T. E. Dunning. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74, 1993.

[8] D. A. Evans, K. Ginther-Webster, M. Hart, R. G. Lefferts, and I. A. Monarch. Automatic indexing using selective NLP and first-order thesauri. In *Proceedings of Intelligent Multimedia Information Retrieval Systems and Management (RIAO'91)*, pages 624–643, 1991.

[9] J. Fagan. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science*, 40(2):115–139, 1989.

[10] T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. Integrating topics and syntax. In *Advances in Neural Information Processing Systems 17*, 2005.

[11] J. Hodges, S. Yie, R. Reighart, and L. Boggess. An automated system that assists in the generation of document indexes. *Natural Language Engineering*, 2(2):137–160, 1996.

[12] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In *Proceedings of the NATO Advanced Study Institute on Learning in graphical models*, pages 105–161, 1998.

[13] J. S. Justeson and S. M. Katz. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27, 1995.

[14] D. J. C. MacKay and L. Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):1–19, 1994.

[15] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[16] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[17] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, 2002.

[18] M. Mitra, C. Buckley, A. Singhal, and C. Cardie. An analysis of statistical and syntactic phrases. In *Proceedings of RIAO-97, 5th International Conference*, pages 200–214, Montreal, CA, 1997.

[19] F. Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19:143–177, 1993.

[20] M. Steyvers and T. Griffiths. Matlab topic modeling toolbox 1.3. http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm, 2005.

[21] T. Strzalkowski. Natural language information retrieval. *Information Processing and Management*, 31(3):397–417, 1995.

[22] H. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[23] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*, 2006.

[24] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information System*, 22(2):179–214, 2004.