Topic-Enhanced Models for Speech Recognition and Retrieval

by

Jonathan Wintrode

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

November, 2015

© Jonathan Wintrode 2015

All rights reserved

Abstract

This thesis aims to examine ways in which topical information can be used to improve recognition and retrieval of spoken documents. We consider the interrelated concepts of locality, repetition, and ‘subject of discourse’ in the context of speech processing applications: speech recognition, speech retrieval, and topic identification of speech. This work demonstrates how supervised and unsupervised models of topics, applicable to any language, can improve accuracy in accessing spoken content.

This work looks at the complementary aspects of *topic information* in lexical content in terms of local context - locality or repetition of word usage - and broad context - the typical ‘subject matter’ definition of a topic. By augmenting speech processing language models with topic information we can demonstrate consistent improvements in performance in a number of metrics. We add locality to bags-of-words topic identification models, we quantify the relationship between topic information and keyword retrieval, and we consider word repetition both in terms of keyword based retrieval and language modeling. Lastly, we combine these concepts and develop joint models of local and broad context via latent topic models.

ABSTRACT

We present a latent topic model framework that treats documents as arising from an underlying topic sequence combined with a cache-based repetition model. We analyze our proposed model *both* for its ability to capture word repetition via the cache and for its suitability as a language model for speech recognition and retrieval. We show this model, augmented with the cache, captures intuitive repetition behavior across languages and exhibits lower perplexity than regular LDA on held out data in multiple languages. Lastly, we show that our joint model improves speech retrieval performance beyond N-grams or latent topics alone, when applied to a term detection task in all languages considered.

Primary Reader: Sanjeev Khudanpur

Secondary Reader: David Yarowsky

Acknowledgments

> A great and glorious thing it is
> To learn, for seven years or so,
> The Lord knows what of that and this,
>   Ere reckoned fit to face the foe.
>
> Rudyard Kipling, ARITHMETIC ON THE FRONTIER

Like so many good things, a dissertation cannot happen in a vacuum, intellectual, professional or personal, and for the past seven years or so countless friends and colleagues have provided support, ideas, feedback, encouragement, and friendship.

To Sanjeev Khudanpur, thank you for your time, ideas, feedback, patience, quite often signatures, and infectious enthusiasm.

To Ben Vandurme and David Yarowsky, thank you for all your insights and willingness to see this thesis through to then end!

To Jack Godfrey for supporting and encouraging me setting out on this path.

To Cathy Thornton, thank you for always being ready to help in navigating any administrative obstacle that might arise.

To Yenda Trmal, Dan Povey, and the rest of the JHU Kaldi team, thank you for

allowing me to be a part of your successes.

To Gypsy Phillips, Jon Nedel, Michelle Fox, and my colleagues in Maryland and across the world, thank you for all of your support, patience, and encouragement over the years.

Merci à tout et pour tout!

Dedication

This thesis is dedicated to my family: my wife Brenda and my children Timothy and Alice; to their love, encouragement, patience, and support. Above all, this is for Brenda, who has always believed in me and has never let me stop believing in myself.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The goal of this thesis is to leverage multiple aspects of topical information in spoken language to improve access to informal media. By richer modeling of topical phenomena in spoken language we aim to improve speech recognition and speech retrieval systems. Our English word *topic*, which captures the abstract notion of a particular 'subject of discourse', arises from the Greek root, $\tau o \pi o \varsigma$, meaning a physical 'place' or 'location'. As the etymology suggests, the semantic concepts of a particular subject are not disjoint from the physical location of the words themselves.

For this reason we focus this work on two related aspects of *topic*, subject-relatedness and locality. First, word usage is affected by the semantic 'subject of discourse' and secondly, word usage is affected by proximity. Two words are topically related because they refer to the same subject, but likewise words are topical in the sense of sharing the same place ($\tau o \pi o \varsigma$).

In this thesis we examine the idea that in modeling informal speech, these two modes of topicality are complementary in the sense that we can leverage them for a joint positive impact on various speech retrieval tasks. We examine both properties of topicality in the context of speech recognition and retrieval and conclude by offering a framework to jointly model both locality and subject-relevance.

1.1 Motivation

Informal spoken content is being generated, stored, and shared on mind-boggling scales across the globe. Smart phones and social media, among other technologies, have enabled the creation of **high volume** repositories of user-generated, **informal** content in almost all **languages**. A recent snapshot from YouTube has users uploading over 100 hours of video every minute, 75% of which is coming from outside the United States and is localized over 60 countries and languages [1].

The problem underlying this thesis is how to organize this wealth of language-rich, *spoken* content and "make it universally accessible and useful" [2]. This touches on many individually challenging application areas such as speech recognition, language modeling, and information retrieval. Three constraining factors are the wide variety of languages, the informal genre of much of the user-generated content, and the massive data volumes.

Because of these limitations on processing high volumes of multimedia in diverse languages, to date little of the content itself is accessible in the same manner as traditional web documents. User tags, links, PageRank, user compiled lists or 'channels', or other metadata are the means by which one links to multimedia content. None of the linguistic content encoded in the audio or video signal is used in the retrieval process.

The diversity of languages implies that in most cases applications operate in languages without extensively annotated corpora on which automated processing algorithms are typically built. Both corpora limitations and data volumes (which imply processing speed and accuracy trade-offs) require operating in an extremely noisy environment, as measured by traditional metrics such as word error rate (WER).

We choose to focus on **topicality** because of the mass of evidence that the topic signal in informal speech is highly robust to speech recognition errors (cf. [3]). We argue that leveraging this robust information is a reasonable route to effective systems in such an environment. Whereas various authors have studied **topicality** in respect to one or more of the aforementioned application areas, we aim to develop a unified approach, focused on speech retrieval as the end goal.

Although online media content covers a broad spectrum from entertaining to informative, we motivate the effort to improve access to all this content with the words of an Egyptian protester in Tahrir Square during the 2011 Arab Spring popular uprising:

> We use Facebook to schedule the protests, Twitter to coordinate, and YouTube to tell the world [4].

Although not all such informal content has the geopolitical import of the Arab Spring protest movement, the ability to access online videos (YouTube), lecture videos (MIT Lecture project), oral histories (the Malach project) online course material (Kahn academy), instructional videos, entertainment, and in a corporate setting, accessing meeting interchanges would benefit a variety of demographics.

1.2 Speech Retrieval

We consider the application of topic information to speech retrieval from the perspective of an information retrieval (IR) task.

> Given the user query, the key goal of an IR system is to retrieve information which might be useful or relevant to the user. [5]

The notion of what "might be useful" is expressed as the user's *information need*. We can argue that one criterion for a document being relevant to the user's query is that the topic of the document, in terms of its 'subject of discourse' contributes information that matches the user's need or answers the user's question. We can think of the notions of information need and topic of interest as synonymous.

In speech retrieval the modality of the relevant documents is different, but the overall goal is the same. In practice, however, we have to transform raw multimedia data into a format that can be indexed and searched efficiently in response to user queries. Typically this transformation is effected by automatic speech recognition (ASR). We will refer to ASR portion of this process as *tokenization* so as to be

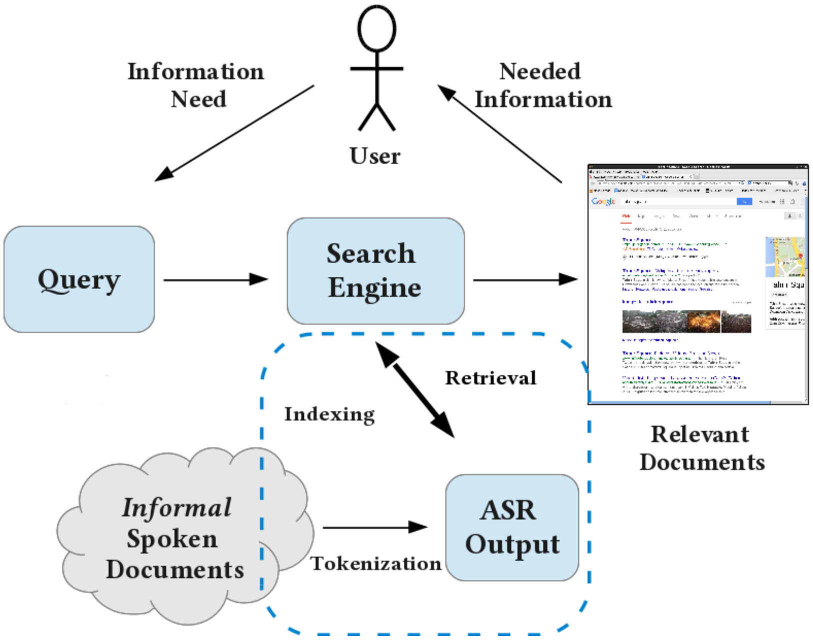


Figure 1.1: A typical speech retrieval workflow.

agnostic to the types of word or other units used to characterize the universe of documents. Figure 1.1 illustrates this stylized view of speech retrieval.

Early attempts at speech retrieval treated the *tokenized* documents as if they were human-generated text documents and applied standard text-based IR systems to the output. When this approach was applied to broadcast-only style media, during the 2000 NIST TREC Spoken Document Retrieval (SDR) evaluation [6], the consensus was that SDR was now a solved problem, given the relatively high accuracy of ASR systems applied to formal, broadcast speech.

However, when NIST revisited the issue in 2006 with the Spoken Term Detection evaluation [7], a different set of conclusions emerged. The 2006 evaluation focused on informal speech and languages other than English (Mandarin Chinese and Levantine Arabic) and on conversational speech in addition to the traditional broadcast news domain. Performance on these other languages were about 50% worse than the English systems. Additionally, by treating ASR output as distinct from plain text, techniques such as indexing multiple ASR hypotheses led to significant gains over the black-box approach from the 2000 TREC eval. [8]

For this reason Figure 1.1 shows the tokenization, indexing, and retrieval steps in the overall workflow broken out explicitly. We would consider the application of topic information to all three areas of the speech retrieval process.

1.3 Topics in Recognition and Retrieval

An additional aspect of 2006 NIST evaluation, the evaluation criteria, suggests that incorporating topic information is a reasonable direction to explore in with respect to extracting information from spoken content a language-rich digital environment. Rather than evaluate speech recognition as a *transcription* task, where accuracy is measured over all words in the corpus - i.e. the word error rate (WER), the 2006 and subsequent evaluations focused on the retrieval of key words and phrases. In other words, we would measure our system accuracy not over all words, but the information-rich 'topic' words.

If we look at model-based retrieval, which arises in the literature as text categorization or classification (e.g., spam filters, document routing, author attribution), we find that most algorithms operate on bags-of-words, which are simply *accumulated token counts*. As a consequence, we need not be constrained by the accuracy of particular token instances - the WER - and can attempt the task with *limited training* higher WER systems.

For this reason we would like to focus on introducing topic information into the retrieval pipeline. We focus on the term detection or keyword search task as our particular instantiation of speech retrieval in keeping with recent evaluations (cf. [7], [9]). For both the tokenization step and for indexing/retrieval we direct our emphasis at adding topic information to the modeling of word sequences: *language modeling*.

For tokenization or ASR, the basic statistical question is to identify the most likely sequence of words given the observed acoustic signal. Also described as the noisy channel model of ASR, we often see this expressed as:

$$\hat{W} = \underset{W}{\operatorname{argmax}}\, P(W|O) \approx \underset{W}{\operatorname{argmax}}\, P(O|W) \cdot P(W) \tag{1.1}$$

We will make a reasonable simplifying assumption that the acoustics of a word, $P(O|W)$, are unrelated to any topic information about a particular word instance. Which again brings our focus to the latter component of a type speech recognizer, the language model $P(W)$.

Similarly, for keyword retrieval, we are interested in the likelihood of the query word or phrase at a particular time instance, which we can also express by the above equation, only without the ‘argmax’. So for both recognition and retrieval we will discuss what topic information can be included in the ***language model***.

Typically, and particularly so for ASR, the language model for a word sequence W of m words, usually denoted as $w_1, \ldots, w_m$, is expressed, via the chain rule, as the product of the individual word probabilities conditioned on a short word history or word *context*.

$$p(W) = \prod_{i=1}^{m} p(w_i|\Phi(w_i)) \tag{1.2}$$

In all major commercial ASR systems this context is expressed assumed to be the $(N-1)$ words immediately preceding w_i, hence the N-gram language model. However, we chose to let $\Phi(w_i)$ stand for any ***context*** that influences the occurrence of w_i - N-grams, syntax, repetitions, or ***topic information***.

The specific goal of this thesis then is to relate the two modes of topic information, subject-relatedness and locality, which we informally refer to as broad and local topic context, to formal language models. Both broad and local contexts influence word usage in language and we show that by modeling word in such a manner improves speech recognition and retrieval tasks.

1.4 Contributions

We aim to analyze the behavior of topic information in informal speech and to model that behavior in ways to improve speech retrieval applications.

Locality for Topic Classification - We will demonstrate that a temporal analysis of the topic signal (locality) can be used to improve topic classification of informal speech.

Locality and Topicality for Speech Retrieval - We will demonstrate that we can model both locality of word usage and subject relevance to improve speech retrieval. We show locality can be expressed implicitly as part of the retrieval task, but also explicitly as part of the language model for the speech recognition component of the retrieval task.

Cauche-augmented Latent Topic Models - We will capture our intuition from the previous two results and describe a latent topic model that incorporates both the subject-relevance aspect of topicality as well as locality or repetition-based properties. We demonstrate that broad and local context, as we have defined them, are complementary sources of information when applied to speech recognition and retrieval.

1.5 Outline

The rest of this thesis is organized as follows. In Chapter 2 we present background materiel placing the notion of 'topic' in context with classification, language modeling, speech recognition and retrieval. We aim to present a concise picture about how different techniques have been used to incorporate topic information into speech and language processing. Chapter 3 examines how topics and location interact in classification-based retrieval of speech. We also show how retrieval metrics provide a better gauge of system error with respect to topic-related tasks than tradition word-level transcription metrics.

In Chapter 4 we present three approaches relating to how topic information both in terms of locality and subject-relevance can be applied to language models and to speech retrieval. We formalize this intuition in Chapter 5 and present a set of locality-aware latent topic models targeted for speech recognition and retrieval. In Chapter 6 we analyze the ability of our proposed models to capture both aspects of topicality and in Chapter 7 we focus on our model's application to the speech retrieval task. Finally, we summarize the individual components and their connection to topicality in speech and discuss possible directions for future work.

Chapter 2

Background

The goal of this chapter will be to highlight the body of research from a range of fields at the intersection of topic and language modeling, speech recognition, and retrieval. In particular we will highlight where topic information, both in terms of subject-relevance and in terms of locality, has been incorporated into various processes, algorithms, and models of speech and language.

We begin by defining a set of commonly used evaluation metrics to which we will refer throughout the rest of this and subsequent chapters. We will then look at the most straightforward application of topic information, document **classification**, with an emphasis on spoken document classification and to highlight the robustness of the topic signal. We also discuss related work in which the locality of information is studied or leveraged.

In Section 2.3 we examine the role of topic and locality as applied to **speech**

recognition. Although much of this work is applicable to language modeling in general, we focus on its impact on speech recognition and related applications. We then discuss how topic and locality have been applied to models for information retrieval, primarily in the text domain.

Finally we examine the connection between different discrete random process formalisms and how different **generative models of language**, such as latent topic models and N-gram language models, arise, and in particular we highlight their different expressions of locality.

2.1 Evaluation Metrics

The identification error rate, classification error rate, or simply **ID Error** is the fraction of incorrect labels applied by the system out of the N total test items:

$$Error \equiv \frac{\#incorrect}{N} \tag{2.1}$$

Related to this is the Word Error Rate (**WER**) of a transcription task, which requires an alignment to the reference transcript in order to count different error types - substitutions (S), insertions (I), and deletions (D). Note that because of the accounting for insertions, errors can outnumber the references words W. Anecdotally, a $WER > 1$ typically indicates an error or bug in the experiment configuration, not an extremely poor performing system.

$$WER \equiv \frac{S+I+D}{W} \tag{2.2}$$

If we look at a system from the point of view of detection - detecting words or documents or topics - a common metric from the Speaker and Language ID communities is the **Equal Error Rate** (EER). By measuring the probability of missing a correct detection, $P(miss)$, and the probability of a false alarm, $P(FA)$, EER is defined as the value at which the two quantities are equal for a particular set of detections.

$$EER \equiv P(miss) = P(FA) \tag{2.3}$$

Specifically for term detection (keyword search) evaluations, NIST defined a Term Weighted Value metric for measuring keyword detection accuracy, for which, unlike the previous three error metrics, higher is better. Also defined in terms of $P(Miss)$ and $P(FA)$, TWV is based on weighted cost function balancing the importance of misses and false alarms. TWV is computed given a fixed score threshold θ, and is averaged over all query terms in some evaluation set Q. For the NIST evaluations, Q is defined explicitly as a list of key words or phrases, but we can think of this as any discrete set of queries.

$$TWV \equiv 1 - \frac{1}{\|Q\|}\sum_{q \in Q}[P(miss, q, \theta) + \beta \cdot P(FA, q, \theta)] \tag{2.4}$$

The cost tradeoff parameter β can be set, in theory, to any value reflecting an application's preference for high recall (low $P(miss)$) or high precision (low $P(FA)$) results.

Lastly, we define related ranked retrieval metrics, typically used by the information retrieval community, but applicable to any scenario in which a ranked (ordered) list of results and binary judgments (correct or relevant) for each result is available. **Recall** and **precision** can be defined, at any threshold in the list, as the number of correct results (C) over either the total number of positive examples in the list (T) or the number of hypotheses in the list before the threshold (H).

$$Recall \equiv \frac{C}{T} \quad Precision \equiv \frac{C}{H} \tag{2.5}$$

Average precision (AP) is found by computing precision for each threshold where a correct item is found in the list. So with T total correct items, AP is computed from T precision values. **Mean average precision** (MAP) is simply average precision computed for each of the queries in the test query set Q. MAP can also be interpreted as the Mean Area Under the recall-precision Curve (MAUC).

Of all the metrics described, **EER** and **average precision** depend only on the rank order of a particular result set. That is to say they do not require calibrated scores or the selection of a particular threshold when computed on the list of results from a particular query or scores from a classifier over a set of documents. We mention this in particular for the **TWV** keyword search metric, which is particularly sensitive to thresholding. In subsequent sections we will make the distinction between techniques that keep system output the same but alter score values (and thus thresholding) versus techniques that cause the system, a speech recognizer, for example to output a fundamentally different set of results.

2.2 Topic Classification

Since the late 1990's there has been an accumulation of evidence supporting the claim that topic classification of speech is highly robust to ASR errors. We use the term *topic classification* to describe a set of tasks also referred to as *topic identification*, *text categorization*, *topic detection*, *topic filtering*, or in a call center context, *call routing*. These techniques may be used for the retrieval task in cases where a user provides examples of the content in which they are interested, where supervised machine learning algorithms are applied to user input.

Two excellent overviews to classification of text and speech can be found in [10] and [11] respectively, but we will briefly describe common relevant assumptions here. Sebastiani describes the basic machine learning problem of text categorization as:

> the task of assigning a Boolean value to each pair $d_j, c_i \in D \times C$, where D is a domain of documents and $C = \{c_1, \ldots, c_{|C|}\}$ is a set of predefined categories. [10]

For our purposes assume 'categories' correspond to 'topics' in the sense of discourse subject. The machine learning problem is then, given N labeled examples (d_j, c_i), to assign the correct label to some new document d. The relationship between classification and word distributions of language arises when we consider how to represent a document d.

Each document d is typically represented by a real-valued vector W where $W = \langle w_1, w_2, ..., w_{|V|} \rangle$. The process of generating W from the lexical content (written or spoken) of d is usually referred to as the *feature extraction* step. The most common feature extraction schema is the *bag-of-words* model. Each document vector has $|V|$ dimensions, one for each word in the system vocabulary V. The values w_i for a document d are computed by weighting the number of occurrences (**counts**) of word i in d. Various weighting schemes have been proposed, some specific to particular classifiers (cf. [12], [13]), but a typical weighting is TF-IDF (term-frequency/inverse-document-frequency) based, where common words that occur in many documents (high DF) are discounted. Irrespective of the weighting scheme, bag-of-words vectors are a sparse representation. A small fraction of vocabulary words occur in any particular document. Bags-of-words are entirely count-based. The order of words or any other location information is discarded in this representation.

The use of bags-of-words may contribute to the robustness of topic classification to speech recognition errors. A standard pipeline for this task applies automatic speech recognition (ASR) to the data, then constructs bags-of-words from the extracted word or subword *tokens* for classification. These bags-of-words are based on *accumulated token counts*, not derived from specifics of individual tokens (at their particular locations). As a consequence, we need not be constrained by the accuracy of particular token instances - i.e. the word error rate (WER). As we will show, we can attempt the task with *limited training* higher WER systems.

Existing approaches to topic classification of speech tend to fall into three different categories based on the type of tokenization used: full vocabulary word-based ASR, subword (phonetic) ASR systems or zero-resource systems. Zero-resource refers here to the lack of in-language, transcribed resources for building supervised acoustic models, not the lack of topic labels or unlabeled acoustic data.

Initial work on the Switchboard corpus [14] by Peskin et al. (cf. [15]), using 44% WER transcripts, demonstrated Topic ID error rates comparable to using human transcripts. The 0.8% error on the 10 topic task was so low that until recently, the task was considered trivial. More recent work, on the 40-topic Fisher English and 25-topic Fisher Spanish corpora give a more complete picture of the relationship between recognition errors and topic classification.

Results from Hazen, Richardson, and Margolis [16] using manual transcripts indicate a more difficult overall classification task than Switchboard, irrespective of

WER. They demonstrated the usefulness of leveraging ASR word lattices for the task, achieving a 9.6% error rate, an improvement over 1-best ASR output, but still higher than the 8.2% human transcript baseline. In this case, unlike the earlier Switchboard results, the impact of ASR errors is not negligible. Sacrificing transcription accuracy for improved decode speed, a necessity for high data volumes, [17] found more significant increases in ID error (from 10% to 19%) as WER reached 47%. Nonetheless, the WER's reported above do not approach the 60-70% WER recognition observed during the first evaluation period of the IARPA Babel/OpenKWS program, for the 10 hour Limited LP training condition [18].

2.2.1 Limited Resource Approaches

The implication of limited linguistic corpora and resources when addressing the language diversity of sites like YouTube with a global reach suggests one of two solutions. Focus on the generation of large informal speech corpora for all of the world's 6000+ languages on par with what is available in English or Spanish, or develop sufficiently accurate and viable technology using only limited linguistic resources. Despite all the rage over Big Data - most of the big data is unsupervised. With respect to supervised resources - transcriptions, lexicons, treebanks, etc. - for most languages speech technologies must make do with small data to start.

We can divide existing low-resource approaches between supervised and unsupervised approaches. A typical supervised approach is to train a phonetic or subword

ASR system. Arguments for subword-based approaches are that they require less training data than large vocabulary systems and avoid the limitations of fixed vocabularies. However, in the Fisher experiments on informal speech, in-language phonetic tokens more than double the classification error rate from from 9.6% to 22.9% [16]. Likewise, cross-lingual phonetic recognition, using a phone recognizer in language X to generate tokens from speech in language Y, the error rate more than doubles again to 53%. Subsequent work showed that applying minimum classification error (MCE) training for feature weighting reduces the classification error on in-language and cross-lingual phonetic tokens to 19.2% and 47.7% respectively [19]. This still represents a significant degradation from a word-based approach.

Unsupervised acoustic modeling techniques aim to discretely tokenize speech without benefit of transcribed training data. From the perspective of topicality, if one learns a stable and consistent set of tokens, one can detect the topic signal regardless of how tokens are labeled. In their work on self-organizing units (SOUs), Siu et al. achieved 45.9% error on Fisher using HMMs with Segmental Gaussian Mixture Models (SGMMs) to discover word like units from 4 hours of English unsupervised training data [20]. This result compares to the cross-language phone tokenization in [19].

The *pseudoterm* approach from Dredze et al. reported 7.5% ID error on the Switchboard task [21]. At a high level a pseudoterm is one instance within a cluster of acoustically similar speech intervals. Work by Carlin et al. considered the viability of

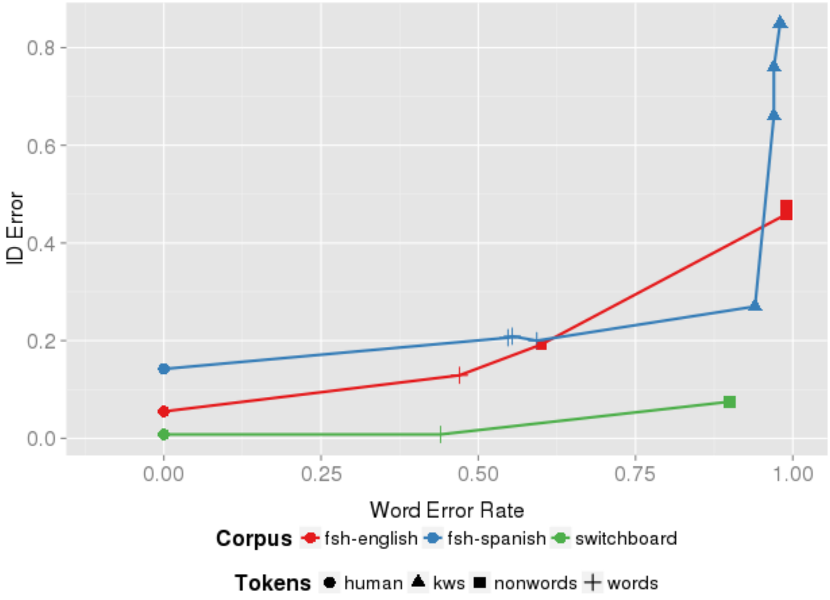


Figure 2.1: Effect of ASR errors on topic classification of informal speech.

various features for pseudoterm discovery during the acoustic match phase, including fully zero-resource features such as FDLP, PLP, and MFCC [22].

With respect to leveraging topicality, we proposed an alternative word-based, low-resource approach using limited vocabulary keyword spotting in [3]. Rather than build full-vocabulary LVCSR systems, we train a keyword spotter on only topic-rich words and generate token counts in a spoken term detection framework. When combined with deep neural net (DNN) acoustic models, this approach achieved Topic ID results near the human transcript baseline. We will analyze these results further in Chapter 3 to consider alternatives to WER in predicting the utility of topic information.

If we collect reported classification error rates on available informal speech corpora (LDC's Fisher English and Spanish, Switchboard), we can plot them against the reported (or estimated[1]) WER (cf. Figure 2.1). Our own experiments on the Fisher Spanish 25-topic task are the most comprehensive, in terms of variety of error conditions and illustrate that the topic information necessary for this particular task resides in at most 10-20% of the word tokens.

2.2.2 Within Document Locality

Much of the literature regarding topic classification of speech focuses on corpora where topic labels are applied at the whole-document level. However, it is realistic to suppose that during an actual conversation, lecture, or other informal spoken document, participants may speak on multiple subjects at various points within the document. Separating a document into coherent topical regions, *topic segmentation* can be considered a task unto itself or useful for downstream retrieval tasks.

Early on in the information retrieval literature, it was recognized that the "subtopic structuring" of documents could be used to improved full-document retrieval [23]. Hearst's TextTiling algorithm [24], used in the aforementioned document retrieval experiments, is the most widely sited text segmentation algorithm in the literature and relies exclusively on bag-of-words "lexical co-occurrence patterns" roughly at the paragraph level. We would argue that her results indicate that information relevant to a particular query is often localized in sub-sections of the document.

[1] We have included only word-based systems in this graph, for which we can compute WER. For word-spotting systems, we estimate WER from keyword detections, treating all other words as out-of-vocabulary.

A number and variety of algorithms have sought to improve on the straightforward sliding-window approach of the TextTiling algorithm. Reynar's Ph.D. thesis refers to the notion of a "topic shift" in developing his segmentation algorithm [25]. Choi's C99 algorithm [26] improves on TextTiling in terms of speed and accuracy.

In terms of segmentation, Bayesian or latent topic models provide a simple framework for expressing this notion of 'subtopic-structuring'. In a Bayesian sense, a topic is defined simply as a distribution over the corpus vocabulary [27]. Given this definition, we can define a document as generated by a weighted mixture of topic distributions. For segmentation, the latent topic distributions of a document vary from region to region within that document. We will discuss latent topic models and their relationship to language models in depth in subsequent sections, but a number of improved topic segmentation models have been developed using Bayesian topic modeling technique.

The current state of the art techniques, Du et al.'s Structured Topic Models [28] and Ngyuen et al.'s SITS (Speaker Identity for Topic Segmentation) model [29] have also been evaluated on informal meeting speech. The ICSI meeting corpus [30] has been annotated to include topic segments. The only other corpus of informal speech, to our knowledge, annotated at this level of granularity is the CallHome Spanish corpus, which was annotated to study discourse structure [31]. However, most corpora are not annotated to this level, so segmentation effects can only be evaluated implicitly.

Given the lack of segment-level labels, there is some work being done to consider the effect of topic locality on the classification task. In our own work, we applied the assumption underlying in Hearst's IR work - that not all document segments need to be relevant to the query - to the classification task. We focused on an aspect of LDC's informal corpora, whereby participants begin discussing the prompted topic, then drift off-topic as the conversation progresses. We found that by modeling this *topic drift* explicitly in a bag-of-words framework, we could reduce the ID error rate by 23-47% [32]. By contrast, in the Reuters text categorization corpus, we found no evidence of topic drift, at least as far as impacted ID error. We will consider these results in more detail in Chapter 3.

Our assumption that the labeled topic in a supervised setting is most prominent at the *beginning* of a spoken document need not be true, and is almost certainly too restrictive in general. Recent work on a "theme identification" task for call centers by Morchid et al. considers a location-dependent model for classification [33]. Here, location is discretized to one of four quantiles of the spoken document and improves classification accuracy by 7% over a comparable bag-of-words system. In this case, no restriction is placed on which quantile is most relevant to the task.

We would draw two main conclusions from the body of work on topic characterization (to include both classification and segmentation) of speech. First, as we have mentioned is the robustness to ASR errors of topic information in terms of the 'subject of discourse'. Second is the weakness of typical bags-of-words models, given the loss of information about word locations. We are certainly not the first to point out the limitations of the bag-of-words assumptions, but we would simply highlight the role of location or proximity in word usage. As we consider other formal models of language for speech recognition and retrieval we will again notice how the locality of word usage must be taken into consideration.

2.3 Speech Recognition

In the context of speech recognition, a number of efforts have been made to augment traditional N-gram language models with topic information. While there is a broader literature focused on the general problem on modeling word sequences, such as incorporating syntax or approaches to N-gram frequency estimation, we highlight efforts on topic and locality in particular.

Two flavors of models have been examined, each focusing on a different aspect of topicality. Cache-based language models (also referred to as adaptive or trigger models) attempt to exploit the 'burstiness' property of language, that is, words are more likely to repeat within the same document. Topic mixture models look to exploit the different word co-occurrence patterns that occur when different topics are discussed within a document. These two areas correspond to our definitions of local and broad topic context, respectively.

Cache-based language models assume that the probability of a word or N-gram in a given word sequence W is influenced both by the global frequency of that word or N-gram as well as the frequency within the current document or preceding K words. The intuition behind this assumption is that most words are rare at a corpus level, but when they occur, they occur in bursts. Because of this, a local frequency estimate, such as from a K word ‘cache’ of recently observed words, may be more reliable than the global frequency. Such a cache or adaptive approach has the advantage of being straightforward to implement. Jelinek [34] and Kuhn [35] both find benefits to using these types of models for speech recognition. Rosenfeld also examined adaptive models within a maximum entropy framework, focusing on what he referred to as ‘trigger pairs’, and also realizing significant gains in WER [36]. More recently, Singh-Miller and Collins adapted Rosenfeld’s work to improve discriminative language models for N-best and lattice rescoring [37].

Adaptive or cache-based language models leverage what is referred to elsewhere as the contagion property of words. Backoff and smoothing techniques for traditional N-gram models have also aimed to model this property, in order to better account for observed word frequency distributions. Arguably the most effective N-gram language model technique, Modified Kneser-Ney smoothing [38], captures this property of language and has proved highly effective for speech recognition. Beeferman et al., building on Rosenfeld’s trigger models developed a model based on expontial families to model the distance between trigger pairs based on the empirical measurements of the strength of this contagion property [39]. Beeferman’s model was initially applied to text segmentation [40] rather than speech recognition. We will discuss the contagion property with respect to language models and its relation to topicality in more detail in the last section of this chapter.

While cache-based or trigger models focus on information within the current document, topic-based mixture models aim to incorporate information based on word usage patterns across documents. The basic idea is to identify in some way the topic or topics in the document to be processed and then to use topic-specific language models in place of or interpolated with a base language model to do the particular computation (decoding, re-scoring, or simple likelihood computation). In some respects, using topic information in this way can considered a form of domain adaptation.

Techniques that attempt to incorporate topics in this manner must first construct a set of topic-dependent language models on training data. This could be done by explicit labels, in supervised setting, or by learning clusters or latent topic models on the training data. Work in 1999 by Florian and Yarwosky [41] and Khudanpur and Wu [42] aim to create topic-dependent N-gram models using a clustering approach. In their work, explicit topic labels were assigned to training to documents via vector-space clustering methods, and then counts taken from the labeled partitions. For speech recognition, first pass output must be used to decide which topic model or models will be interpolated with the global N-gram model. This approach was shown to decrease WER by up to 1% absolute.

Other approaches, using LDA [43], PLSA [44], or similar latent variable mixture models have been proposed similar to the works by Florian and Khudanpur. However, instead of vector-space clustering, topic inference is done within a probabilistic framework, either at training or decoding time. Both Heidel [45] and Hsu [46] use latent topic mixture models to re-score N-best hypotheses using a mixture of topic-dependent and topic-independent N-gram language models based on the inferred topic distribution of the test document. In Hsu's work, however, a hard clustering of documents, rather than latent states, is used when training topic-dependent N-gram models. The work by Hsu resulted in a 2.4% reduction in WER on the MIT Lectures data set. Similar (though smaller) gains were observed by Liu et al. on Mandarin broadcast conversation [47] and by Huang et al. on the AMI meeting corpus [48].

Almost exclusively, the works cited above focus only on *re-scoring* recognizer output. However, the lattice of possible word sequences (and by implication any N-best list) are generated and pruned using the original acoustic and language models. If a word sequence that is more likely under the topic models gets pruned before re-scoring, having a good topic-dependent model does not help.

In the context of latent topic models, we can explicitly define a topic-dependent unigram language model for any given document d, once we have inferred, through any technique, the mixture of topics for that document, $\theta^{(d)}$. For a latent topic model with T topics, $\theta^{(d)}$ is a T-dimensional vector where each element i is the fraction of d that can be attributed to topic i. Practically, $\theta^{(d)}$ acts a mixture weight so that the unigram language model associated with d is given as:

$$P(w) = \sum_{i=1}^{T} \theta_i^{(d)} \cdot P(w|topic_i) \tag{2.6}$$

In recent work, we found that interpolating these document-specific unigram models with the base N-gram model and using this topic-biased model for decoding did indeed have a significant effect on the pruning of keywords. We took four languages from the first and second evaluation periods of the IARPA Babel program [9] (Tagalog, Vietnamese, Zulu, Tamil) and learned an LDA topic model with 100 topics from the training transcripts. We used lattice soft counts from the first pass recognition output in a manner similar to [47] to infer the topic proportions $\theta^{(d)}$ of each document. From this we could construct topic-biased unigram models for each document which we applied during a second decoding pass.

We verified the impact of topic information on lattice pruning by looking at lattice recall of the Babel evaluation keyword lists (roughly 2-5K words or phrases per language). Table 2.1 shows the impact of applying topical unigrams at decoding time, versus the baseline output, measured in terms of the proportion of keyword hits that could be found somewhere in the output lattices. By merely adding topic-dependent unigrams to the base language model, we were able to preserve 2-5% of keyword hits from pruning [49].

Language	**Baseline / Re-score**	**Re-decode**
Tagalog	0.778	**0.792**
Vietnamese	0.555	**0.567**
Zulu	0.718	**0.739**
Tamil	0.573	**0.622**

Table 2.1: Keyword Recall obtained re-decoding with topic-augmented models.

2.4 Retrieval

Language models have applications not just for decoding, but also during retrieval of documents. Language model-based retrieval is major area in the information retrieval (IR) community, staring with the work in the late 90's by Ponte and Croft [50]. Incorporating topicality into the retrieval language models, as with Heart's use of locality, has improved retrieval performance.

In many text retrieval tasks, queries are often tens or hundreds of words in length rather than short spoken phrases. The likelihood of the query word sequence is then evaluated under a document-specific language model. Similar to the manner in which we computed a topic-dependent model from the per-document topic proportions θ_d, multiple efforts in the IR community have used LDA or similar techniques to compute a topic-dependent document model (cf. [51], [52], [53]). In these efforts, the topic model information was helpful in boosting retrieval performance above the baseline vector space or N-gram models.

As previously discussed, topicality also arises in the **burstiness** property of language. Church and Gale examine this property in great depth in their studies of

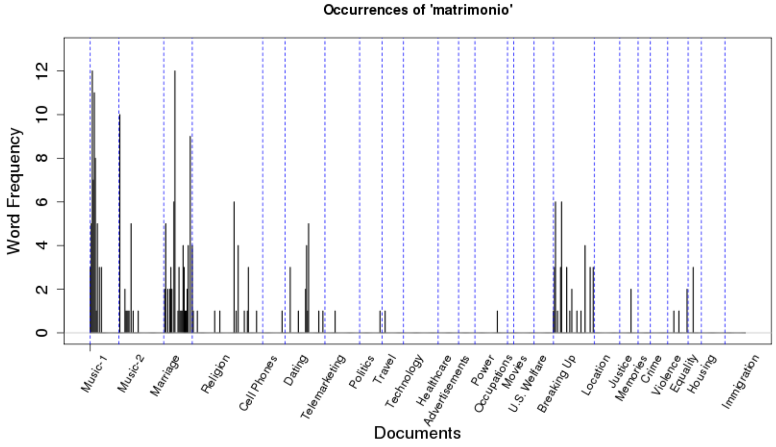


Figure 2.2: Word frequencies of *matrimonio* (marriage) in documents grouped together by topic.

document frequency [54], Poisson mixtures, [55], and word adaptation [56]. Looking at content words in the Brown corpus, they illustrate how content words tend to not be evenly distributed across a corpus (as a Poisson generative assumption would predict), but instead occur in bursts in a small number of documents or topics (genres) [55].

As an example of this phenomenon, we can visualize these bursts on LDC's Fisher Spanish corpus (as Church and Gale did for the Brown corpus) by plotting the frequency of content words in each document, grouping documents with the same topic label adjacent to one another. Documents are given as points on the horizontal axis and dashed blue lines indicate topic boundaries. We can compare the plot for *matrimonio* (marriage, cf. Figure 2.2) with the same plot for a more common word *juntos* (together, cf. Figure 2.3).

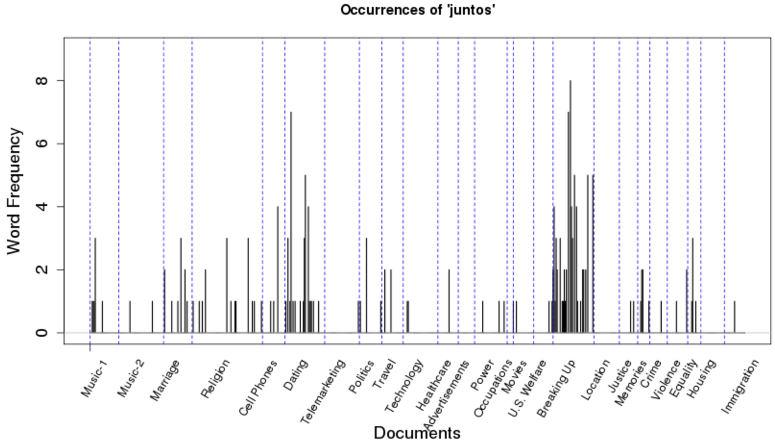


Figure 2.3: Word frequencies of *juntos* (together) in documents grouped together by topic.

While *juntos* is commonly used in contexts unrelated to human relationships, there are noticeable bursts in the *Dating* and *Breaking Up* topics. Taking the χ^2 statistic, used for selecting strongly correlated features for text classification (cf. [57]), we obtain scores of 46.2 and 323.4 for *juntos* co-occurring with those two topics, respectively. For 25 topics, a χ^2 of greater than 44.3 indicates a 99% confidence of a correlation between the frequency of *juntos* and a given topic label. Conversely, the scores for *juntos* and the other 23 topics are less than 11.5, which is the 1% confidence level, as a score of 0 indicates no correlation. In these cases, we are comfortable relating the burstiness in particular documents to the underlying topic.

By contrast, the function words *el* and *como* - ‘the’ and ‘how’ respectively - while quite variable in the frequency with which they occur in particular documents, are not as clearly correlated with particular topics (cf. Figures 2.4,2.5). The highest

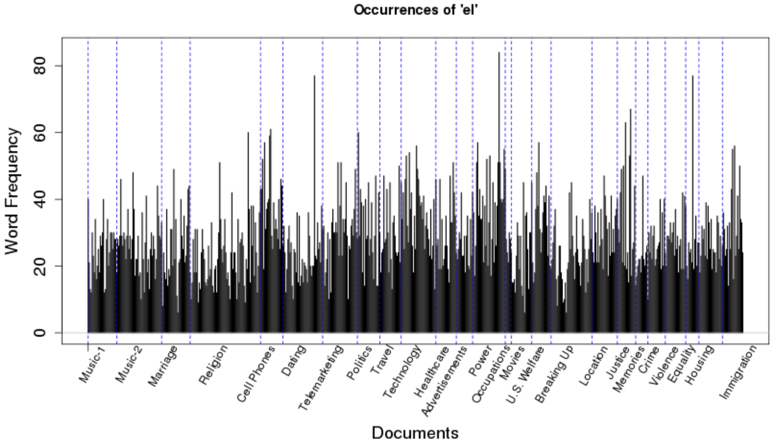


Figure 2.4: Word frequencies of *el* (the) in documents grouped together by topic.

correlated topic for *como* according to the χ^2 metric is *Memories* with a score of 21.4, which is outside the confidence interval. However, the word *el*, according to the χ^2 metric, is correlated with the *Power* topic with a score of 122.0. The word *el* is clearly repeated quite frequently in all documents, but it seems hard to argue that this is because it is closely related to the subject matter.

We will look at the application of both burstiness (or local context) and topic information (broad context) to the retrieval task in Chapter 4. The burstiness property (or contagion) can account for the power law distribution of word frequencies in a corpus. With this in mind, we conclude this discussion of topics within research areas surrounding speech recognition and retrieval by looking at how topics arise in different formal frameworks for statistical language modeling.

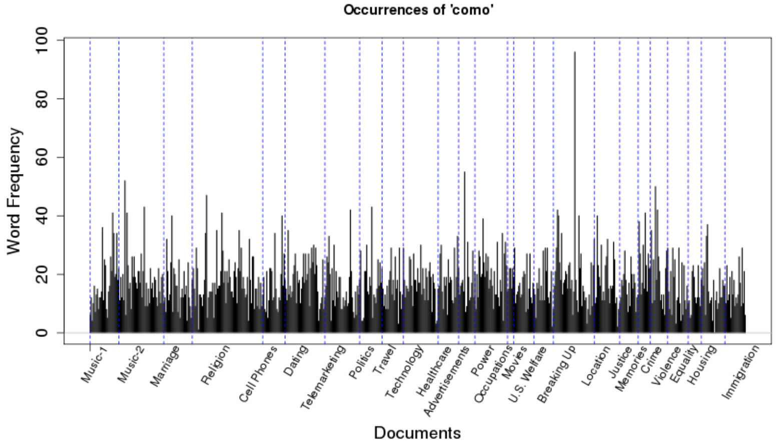


Figure 2.5: Word frequencies of *como* (how) in documents grouped together by topic.

2.5 Generative Models of Language

We have heretofore discussed language models and latent topic models (LDA) somewhat interchangeably. In this section we want to sketch the relationship between N-gram language models and latent topic models, focusing on the burstiness or contagion property. Given the properties of these models, we would then propose our own building on the strengths and weaknesses for our stated task.

2.5.1 Urn Models

There is a family of models, mostly attributed to George Pólya, which can be used to model the burstiness phenomena in language. In the multivariate formulation, we consider an urn filled with balls of V different colors, initially containing x_i balls of color i. At each point in time one is drawn, its color noted. This ball is then replaced along with an additional c_i balls of the same color. This c_i is often referred to as the contagion parameter.

Thus 'burstiness' is modeled as a rich-get-richer scenario. If we use words tokens instead of balls and word types in place of colors, we have a simple generative model for corpora. If we have initial counts x_i and let n_i be the number of additional words of type i that have been drawn, the probability of a particular word at this time in the generative process is given as:

$$P(color_i) = \frac{x_i + n_i \cdot c_i}{\sum_{j=1}^{V} (x_j + n_j \cdot c_j)} \tag{2.7}$$

One of the properties of Pólya urn models is *exchangeabilty*: the probability of an n-length sequence of draws depends only on the number of balls drawn of each color, not on the order in which they are drawn. This property is also evident in bag-of-word models, which represent only counts of words and not their order.

Pólya urn models arise in the related distributions in latent topic modeling: multinomials (the topics) and the Dirichlet (the prior on the multinomials). At the limit, the proportion of balls in the urn scheme is distributed as a Dirichlet with parameters $(c_1, \ldots, c_V)$ identical to the contagion parameters on the urn [58]. Multiple authors also note that the Dirichlet-Multinomial is a Pólya distribution, arising from such an urn scheme (cf. [59], [60], [61]). In a Bayesian setting such as Latent Dirichlet Allocation (LDA), the hyperparameter β for the Dirichlet prior of the multinomial topic word distribution is viewed as a smoothing constant or concentration parameter [27]. In the Bayesian sense this is intuitive, given a V-dimensional Dirichlet acts as a distribution over $V-$dimensional multinomials. But in terms of an urn model, β is simply a uniform contagion parameter.

We can characterize the relationship between latent topic models like LDA and these Dirichlet-Multinomial urn models as many to one. The LDA generative model holds that a document is generated from a weighted mixture of K topics (i.e. Dirichlet-Multinomial unigram models). In contrast, a cache or adaptive language model is a single constrained urn model, where the initial contents of the urn are given by the base N-gram language model, and the contagion parameters are captured by the interpolation weight.

2.5.2 Dirichlet Processes

Urn models can also be thought as arising out of a class of models called Dirichlet processes. Dirichlet processes are thought of as "distributions of distributions." [62] In particular, Pittman-Yor processes, which are a particular family of Dirichlet processes, have been shown to relate to both N-gram language models and to Dirichlet-Multinomial mixtures (i.e. LDA).

Both Goldwater et al. [63] and Teh [64] demonstrate the equivalence between Interpolated Kneser-Ney (IKN) language models and what Teh calls "hierarchical Pittman-Yor" language models and what Goldwater et al. refer to as a "two stage"

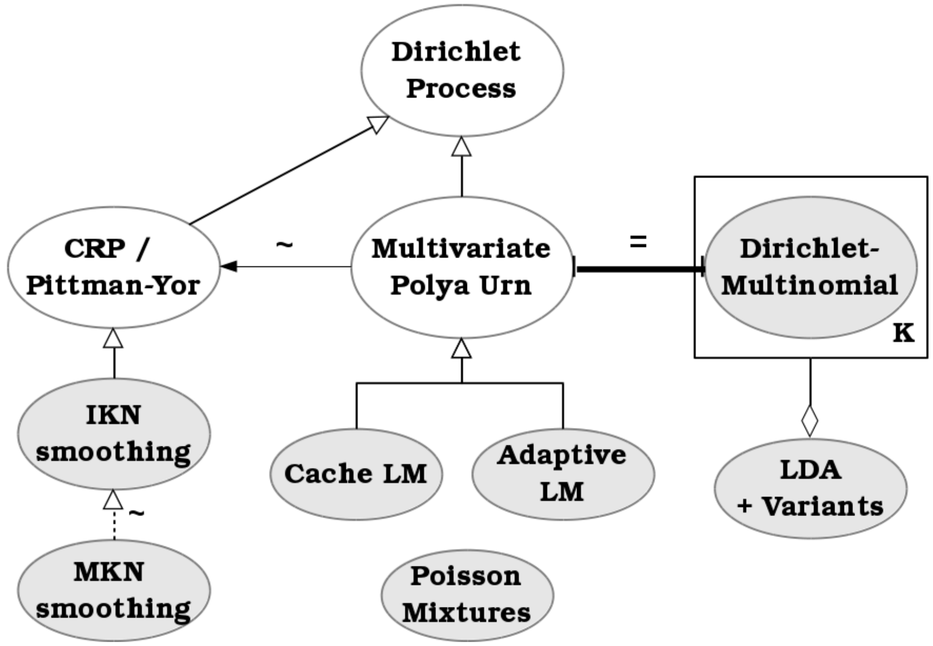


Figure 2.6: Relationships between various models of word generation.

Pittman-Yor process. Teh also argues that Modified Kneser-Ney (MKN) is an approximation to this hierarchical Bayesian non-parametric model.

A stylized UML diagram of the relationship between various language and topic models, based on the previous discussion and showing direct generalization where possible, is presented in Figure 2.6. This is meant to be illustrative, not authoritative or exhaustive.

One final note contrasting MKN or IKN with a cache or adaptive language model, which is directly related to our work in this area. At recognition time, with a typical N-gram language model, the urn is fixed, so to speak. A cache model on the other hand allows the generative process to continue and recognition to benefit from the topic information in the unseen document (at a cost).

In Dirichlet-Multinomial mixture models such as LDA and variants or any the latent variable models such as Du et al.'s Structured Topic Models for example, one of the principle modeling challenges is estimating the unobserved (latent) parameters from the observed data (word sequences). In the literature, this task is referred to as *approximate posterior inference*, and as topic information is often expressed as a latent property of the data we will introduce the most common techniques here. We will refer back to these techniques in Chapter 5 when we develop our own cache-augmented latent topic model.

2.6 Posterior Inference

We now describe the most common posterior inference techniques. In the context of speech retrieval we will distinguish *estimation*, whereby we learn model parameters on some training corpus, and *inference*, where we obtain the latent variable state on unseen or held-out data (i.e. the ASR development or test set). Nonetheless, both steps are examples of approximate posterior inference, in which the latent variable properties are learned from observed data.

Posterior inference techniques typically applied to graphical models can be categorized as Variational Bayes (VB) or Markov Chain Monte Carlo (MCMC) techniques. Variational Bayes techniques perform optimization on a distribution similar to but simpler than the true posterior distribution, typically via the Expectation

Maximization (EM) algorithm. MCMC techniques estimate the posterior distribution by integrating sampled values from a Markov chain that converges to the desired distribution.

Blei and Jordan, in their original derivation of LDA [43], presented a Variational Bayes method for estimating the LDA parameters. This is built off of Jordan's early work where he introduces variational methods that leverage Jensen's inequality to bound the log likelihood [65]. In their original paper, Blei and Jordan refer to the topic distributions with the variable β, and the variational approximation as ϕ[2]. The variational approximation to the topic mixtures θ is given as γ. Blei and Jordan show that the Kullback-Leibler divergence between the true posterior distribution (conditioned on the true parameters β, θ) and the variational distribution $q(\theta, z|\gamma, \phi)$ can indeed be optimized using the EM algorithm.

$$(\gamma^*, \phi^*) = \underset{(\gamma,\phi)}{\arg\min}\, \mathrm{D}\left[q(\theta, z|\gamma, \phi) \| p(\theta, z|w, \alpha, \beta)\right] \tag{2.8}$$

The variational approach necessitates finding some appropriate q that is tractable for optimization techniques. The second approach to posterior inference is a set of sampling techniques referred to as Markov Chain Monte Carlo. MCMC sampling was first introduced by Metropolis et al. [68] in their work on modeling behavior of atomic particles and generalized in a statistical framework by Hastings in 1970 [69].

[2] Except for here, will use ϕ to refer to the original topic distributions, to be consistent with the nomenclature of Stevyvers et al. [66] and related work

The basic idea is arrived at by combining the definition of the expected value of a continuous random variable and Monte Carlo approximations of integrals. The expected value $E[f(X)]$ is given by the integral:

$$E\left[f(X)\right] = \int f(X) \cdot P(X)\, dX \tag{2.9}$$

The intuition behind MCMC is to apply Monte Carlo methods for numerically approximating the expected value of X.

In the specific case of approximate inference, the X we are interested in is just the parameter values given the observed data W (for simplicity, we denote all of our latent parameters $(Z, K, \Theta, \Phi) = X$:

$$E\left[X|W; \alpha, \beta, \nu\right] = \int X \cdot P(X|W; \alpha, \beta, \nu)\, dX \tag{2.10}$$

If one could sample from the posterior distribution $P(X|W)$, then one can numerically approximate the interval in Equation 2.10 and obtain an estimate of the true value. The Markov Chain part of MCMC requires one to produce a Markov chain such that as the sampling procedure progresses, the sample approaches (in the limit) a random sample from $P(X|W)$ [70]. By extension, averaging samples from the chain gives the desired expected value estimate.

Two related MCMC sampling techniques that produce such a Markov Chain are **Gibbs Sampling** and the **Metropolis-Hastings** algorithm. Gibbs Sampling proceeds by sampling one unobserved variable at a time, conditioned on the observed data *and* the current sampler state for the other unobserved values. By the sampler state, we mean the currently sampled values temporarily assigned to the unobserved variables. In our model this would mean sampling each $z_{d,i}$ and then each $k_{d,i}$ given the observed words and current state of the other sampled variables Z and K. The Markov Chain properties of the procedure ensure that as we proceed, the sampled values approach the expected values of the topic and cache states given the observed words. In layman's terms, the expected value gives us the best approximation for the topic and cache mixture underlying the observed words.

In the general case, assuming a series of latent states $Z = \{z_1, \ldots, z_n\}$ and observations $X = \{x_1, \ldots, x_m\}$, Gibbs sampling proceeds as follows. Assign some initial values to the states Z. Then, iteratively, for each z_i, sample a new value for z_i according to the distribution $P(z_i|Z_{-i}, X)$. As elsewhere, the subscript $-i$ indicates that the sequence does not contain the i^{th} item.

Gibbs Sampling can be shown to be a specific case of the Metropolis-Hastings algorithm (see [70] for a full overview of the various MCMC techniques). Metropolis-Hastings constructs the Markov Chain of samples for the unobserved variables by means of *proposal distribution* (also called a *jumping distribution*). The proposal distribution is used to suggest samples, given the current sampling state. Any function $f(Z, X)$ which is proportional to the posterior distribution $P(Z|X)$ we are trying to approximate is used to accept or reject proposed samples, making Metropolis-Hastings a form of *generalized rejection sampling.*

If the current sampling state is Z^t and a new value z' is proposed for z_i, then with Metropolis-Hastings, the acceptance probability α of that particular sample is calculated as:

$$\alpha = \min\left\{1, \frac{f(z', Z^t_{-i}, X)}{f(Z^t, X)}\right\} \tag{2.11}$$

The new sampling value is assigned to the sampling state for z_i with probability α.

In the Gibbs Sampling instance of Metropolis-Hastings, the proposal distribution is simply the distribution $P(z_i|Z_{-i}, X)$ and samples are always accepted. In practice, this need not be the case. A *burn-in* period may be used, where samples are dropped from the first N iterations, so as to allow the Markov process to move away from the initial conditions to a steady state. Another alternative is to use *thinning* where only a proportion of samples are kept. The effectiveness of either technique is best judged empirically.

2.7 Summary

We have aimed to highlight both the multiple aspects of topicality in language, the theoretical frameworks within which they live and by which they are evaluated, as well as the their successful application to various areas of speech recognition and retrieval. Given the positive impact of topic information in its various forms discussed in this chapter, we believe it worthwhile to examine this in more depth.

Chapter 3

Topic Classification

In this chapter we examine locality and topic information in the context of topic classification of speech. We wish to highlight the importance of locality as pertains to the topics of informal speech but also to highlight the error robustness of the topic signal. We demonstrate a much stronger relationship between keyword retrieval metrics and Topic ID Error than between that and word error rate (WER). This analysis leads us to focus in subsequent chapters on more specific models of topic information for keyword-based retrieval.

3.1 Locality in Informal Speech

We first examine the location-sensitive nature of topic information in spoken documents. By a simple initial experiment we can show that information corresponding to the topic label for the available topic-annotated spoken corpora tends to be concentrated towards the beginning of documents (cf. Table 3.1). Building on this result we then present a model for topic locality within a bag-of-words framework. Applying our model to the classification task, effectively ignoring irrelevant words later in documents we can reduce the error rate by up to 50% (cf. Figure 3.1).

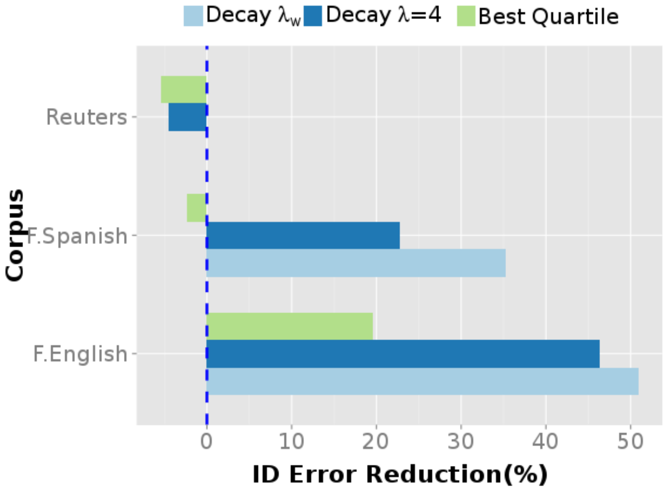


Figure 3.1: Topic ID Error reduction for location-dependent features

Presumably topic location dynamics is domain-specific, so we contrast informal speech with newswire text, where we do not expect our assumptions about speech to hold. We perform the classification task using the Fisher English and Spanish human transcripts and the Reuters RCV1 text categorization corpus [72].

Rather than building bags-of-words from each document in its entirety, we restrict each document vector to a specific quartile. If the topic signal is evenly distributed, we do not expect the performance on one quartile to be significantly better or worse than another. By the first quartile, for example we mean that will construct features

Corpus	All	0-25%	25-50%	50-75%	75-100%
Reuters	**20.1**	23.5	21.9	21.7	21.2
Fisher English	11.2	**8.9**	24.3	38.5	43.6
Fisher Spanish	**25.0**	25.6	34.7	42.1	48.3

Table 3.1: ID Error rate (%) observed when training and testing on data by quartile.

for both training and testing classifiers from only the first 25% of words (by position) in each document. So for a document with 100 words, we would only count words 1 to 25 for the first quartile test, words 26-50 for the second 25%, and so on.

Table 3.1 shows the results of this simple test using a Naive Bayes classifier as described in [32]. Not surprisingly, performance on the Fisher corpora, in which participants are asked to call in and are prompted to talk about a particular topic (which is used as ground truth for the task), is significantly higher on the first quartile, roughly as good as using the entire conversation, and significantly lower on the remaining 75%. Conversely, the Reuters corpus does not exhibit any obvious location dependence, as we expected.

3.1.1 Static Topic Drift

In the general case, there is no reason to believe an arbitrary division such as a quarter of a document should capture within-document topic locality. Instead we propose to model the observed *topic drift*, in which the contribution of individual occurrences of words to the effective topic signal (as measured on the identification task) decreases over the course of the document.

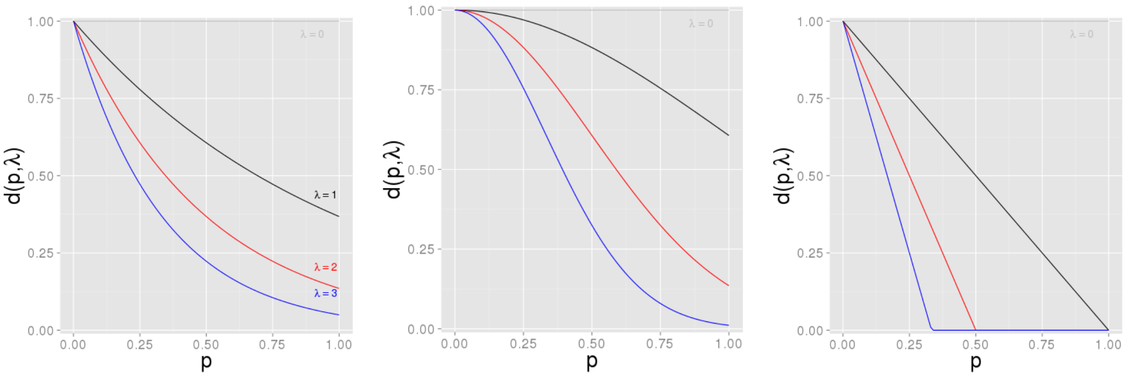
We first model drift by supposing a global decay rate for all words in the vocabulary. This idea is similar in spirit to decaying cache language models (cf. [73]). Within our framework we can also learn word-specific decay rates for each word using a minimum classification error (MCE) training framework. With the dynamic approach, for the informal corpora we are able to cut ID errors in half from the full bag-of-words baseline.

To apply a decay model to the word counts of a document when computing a bag-of-words vector, for each word type we apply a decay function to each token instance evaluated at position p and with decay rate λ and sum over tokens. So the count c_w for a word w in a document of $|D|$ words is given as:

$$c_w = \sum_{i=1}^{|D|} d\left(p = \frac{i}{|D|}, \lambda\right) \cdot I_w(w_i) \tag{3.1}$$

$I_w(w_i)$ is an indicator function whose value is 1 where $w_i = w$ and 0 otherwise.

We considered three possible decay functions, exponential, gaussian, and linear (cf. Eqns. 3.2,3.3,3.4) over the range $[0, 1]$. Examples of each for different λ values



(a) Exponential (b) Gaussian (c) Linear

Figure 3.2: Decay function behavior for selected λ.

are shown in Figure 3.2.

$$d_{exp}(p,\lambda) = exp\left(-\lambda \cdot p\right) \tag{3.2}$$

$$d_{gauss}(p,\lambda) = exp\left(-\frac{\lambda^2 \cdot p^2}{2}\right) \tag{3.3}$$

$$d_{lin}(p,\lambda) = \begin{cases} 1 - \lambda \cdot p & : p \in [0,1] \\ 0 & : p \notin [0,1] \end{cases} \tag{3.4}$$

In the static case, we swept values for λ from 0 to 5, where 0 corresponds in each case to the unweighted baseline counts and 5 being the point after which we observed no additional benefit or degradation in performance. The linear decay generally performed poorly, regardless of λ.

The best results of the static model are listed in Table 3.2. Again we see that modeling drift has no significant effect on performance on the Reuters text corpus whereas for the informal speech, the exponential decay with $\lambda = 4$ decreases ID error over the bag-of-words baseline by 46% and 23% relative for English and Spanish respectively. Interestingly enough, evaluating $d_{exp}(p = 0.25, \lambda = 4) = 0.37$ shows that words after the first quartile are discounted by at least 63%, but do contribute a small amount to the overall weighted word counts.

3.1.2 Word-specific Topic Drift

While the static model is effective in applying location-dependent weighting to all words uniformly, our dynamic model supposes that specific words are more or less sensitive to their position in the document with respect to the Topic ID task. We simply extend Equation 3.1 with per-word decay rate λ_w.

$$c_w = \sum_{i=1}^{|D|} d\left(p = \frac{i}{|D|}, \lambda_w\right) \cdot I_w(w_i) \tag{3.5}$$

We optimize per-word weights λ_w using a Minimum Classification Error (MCE) discriminative framework. We configure our Naive Bayes classifier to compute a score $S(t|D)$ for each document D against each topic t. We can compute a loss function based on a misclassification measure $M(D)$ and maximize the difference between the score for the correct topic t_C and the highest scoring incorrect hypothesis t_I.

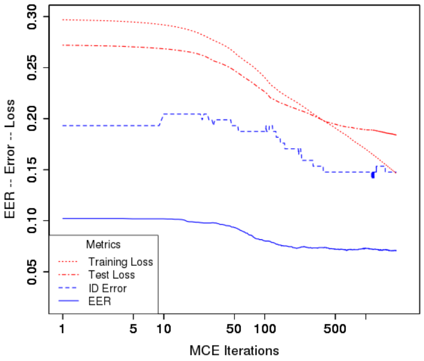
$$M(D) = S(t_I|D) - S(t_C|D) \tag{3.6}$$

We compute the partial derivative and update equations for a gradient-descent optimization of $M(D)$ over λ_w. These partial derivatives, with the score function $S(t|D)$ given by a Naive Bayes classifier based on our modified counts c_w, can be expressed terms of the decay functions. A full derivation for λ_w updates is provided for reference in Appendix A.

Allowing the gradient descent to run to convergence reduces the error rate further over the static model. the overall results are listed in Table 3.2. Learning λ_w for an exponential decay (d_{exp}), outperformed the best static decay model, which was also exponential, with a fixed decay reate of $\lambda = 4$. These numbers correspond to the error rate reductions highlighted a the start of the chapter (Figure 3.1). A representative run of the MCE training, contrasting the loss function on both train and test data with the observed error metrics on the test data is given in Figure 3.3a.

If we look at the words with the highest learned decay rate (Table 3.3b) on the Spanish corpus, two things stand out. High decay rates are learned on both information rich words (*guatemala, méjico, boston*) and typical stopwords (*hm, oh, sí*). Given the corpus, where participants who do not know one another call into a switchboard at LDC to be recorded, this makes sense. The place names, while information rich, typically occur during the chit-chat at the beginning of the conversation, but are in fact irrelevant to the topic.

As we have seen, topic information can be highly localized, and we argue that the phenomenon we have observed and modeled lend support to the consideration of



(a) Error measures on learning λ_w.

Word (Decay Rate)			
hm	10.44	boston	8.31
guatemala	10.19	muy	8.28
asunción	9.95	más	8.16
acá	9.85	chicago	8.13
oh	9.09	miércoles	8.09
sí	9.01	puerto	8.06
uh	8.86	me	8.04
méjico	8.66	um	7.96
ajá	8.52	es	7.86
bonito	8.40	uhum	7.83

(b) Highest decay-rate words

Figure 3.3: MCE training measures

Corpus	λ	**Iterations**	Error	
			d_{gauss}	d_{exp}
Reuters	0	-	-	20.1%
	1	-	20.5	20.5%
English	0	-		11.2%
	4	-	7.3%	6.0%
	λ_w	2000	5.7%	5.5%
Spanish	0	-	-	25.0%
	4	-	19.9%	19.3%
	λ_w	2000	15.3%	14.2%

Table 3.2: Topic ID error by corpus for feature decay models.

decaying cache style language models. We will now turn our attention to topicality and its relation to speech recognition errors. Although topics are highly localized, the information is extremely robust to a high number of ASR errors.

3.2 Word Error Robustness

As we discussed in Chapter 2, topic classification performance on speech recognition output is remarkably insensitive to changes in word error rate (WER) over a broad range of reasonable operating points.

In the rest of this chapter we present and elaborate on work originally presented in [3] to demonstrate how errors on information-rich keywords are more predictive of classification performance than WER. These experiments motivate subsequent chapters where we focus on using topic information to target improvement of KWS accuracy.

The reason why classification performance is insensitive to WER is fairly intuitive, particularly given the work on feature selection for classification (cf. [57], [16]). Feature selection experiments generally indicate that using only the most discriminative words results in equal or better performance than using the entire vocabulary. WER by contrast is computed over all words in the vocabulary, as we mentioned in Chapter 2. Feature selection results imply that most words are uniformly distributed between positive and negative training examples (and thus uncorrelated with the topic label). We would assume that the insensitivity of classification error to WER changes indicates that errors on these 'uninteresting' words are also evenly distributed between positive and negative training examples. In other words, most of the change in WER is related to these words that have no relevance to the topic content.

We replicated the analysis in [57] and demonstrate this effect on the Fisher Spanish

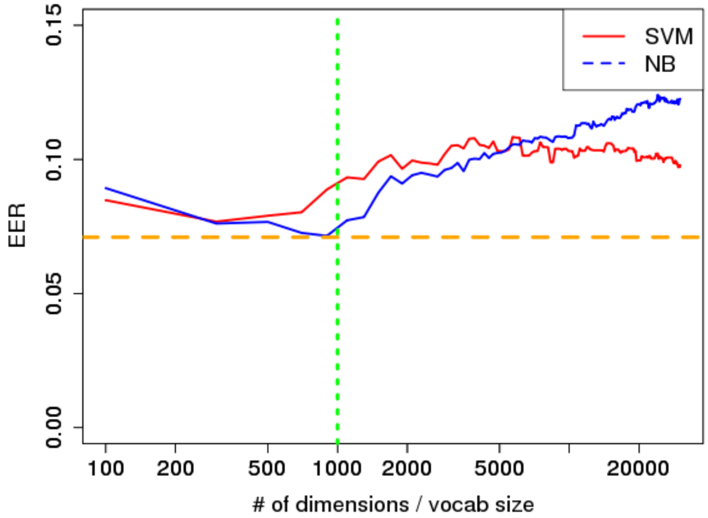


Figure 3.4: Effect of χ^2 feature selection on the Fisher Spanish classification task.

25-topic classification task described previously. By selectively increasing the vocabulary used for classification based on the χ^2 statistic, we achieve the best error rates using only a small fraction (2-3%) of the vocabulary (Figure 3.4). We test a wider range of error conditions using keyword search metrics, by narrowing our focus to the top 1000 words, according to the χ^2 statistic.

3.2.1 ASR Models

To capture the relationship between ASR errors and topic classification performance, we first decode the Fisher Spanish with a range of acoustic and language models of varying complexity. We contrast the performance of the actual ASR system with randomly generated word errors induced on the ground truth but covering a wider range. To increase the amount of error further (without simulation and using the same training data) we also construct limited vocabulary keyword spotters for the same task.

We first use the Kaldi speech recognition toolkit [74] to train a 45K word Spanish ASR system on only the 14 hour Spanish Call Home data [75]. The vocabulary and pronunciations are also restricted to the Call Home lexicon. This translates to out-of-vocabulary (OOV) rates of 47% (types) and 5.7% (tokens). In fact, roughly 10% of our top keywords were OOV.

For all acoustic model training, we use 13-dimensional perceptual linear predictive (PLP) features. These PLPs are used to train both speaker-independent and speaker-adapted triphone models using typical state-clustered HMM's with GMM output densities (denoted **GMM** in subsequent figures). We also trained Subspace GMM's [76] (SGMM) from the PLP's as well (denoted **SGMM**). The SGMM parameters can also be boosted with a maximum mutual information (MMI) criterion. All models use a trigram language model estimated on the Call Home transcripts, and the individual data points for a specific acoustic model reflect varying the language model weight during decoding (cf. Figure 3.5).

We also use Kaldi's CPU-based deep neural net (DNN) acoustic models in a hybrid HMM-DNN configuration [77] (denoted **DNN**). For small training sets ($\sim$10hrs) Kaldi uses a smaller network configuration of only 2 hidden layers and 879 input and output dimensions. The DNN features had little impact on the full vocabulary ASR results, but large impact on the keyword spotting results, as we will see.

Figure 3.5 shows just how little effect a large change in WER has on topic classification, consistent with previous work. The dashed lines indicate the EER of topic classification training and testing on human transcripts with an SVM (7.1%) or Naive Bayes (12.2%) classifier. We see that a 9.4% difference in WER still falls within the bounds of performance on manual transcripts.

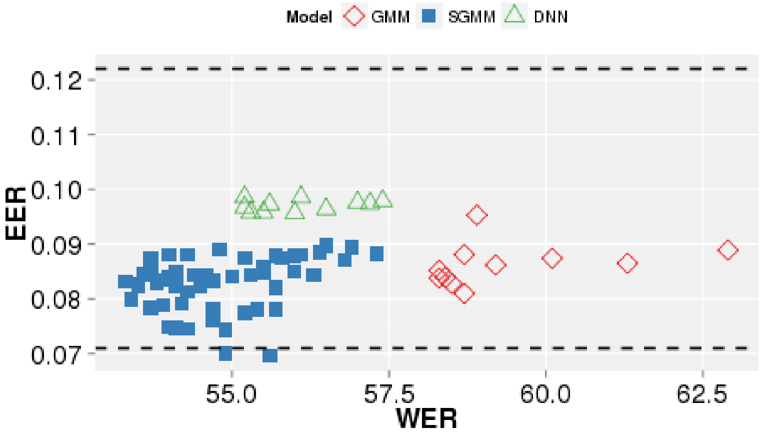


Figure 3.5: Relation between WER and topic detection EER for Fisher Spanish

To understand the significance of this performance range we simulate word errors over the entire range from 5 to 95% WER by randomly inducing either substitutions or deletions in the ground truth transcripts. The full vocabulary systems tended to exhibit roughly twice as many substitution errors as deletions. To induce a 35% WER system, each word in the true transcripts had a 35% change of being modified, and if chosen, a 33.3% change of being deleted or a 67% chance of being replaced with an incorrect word. Word substitutions were selected uniformly from the vocabulary. In

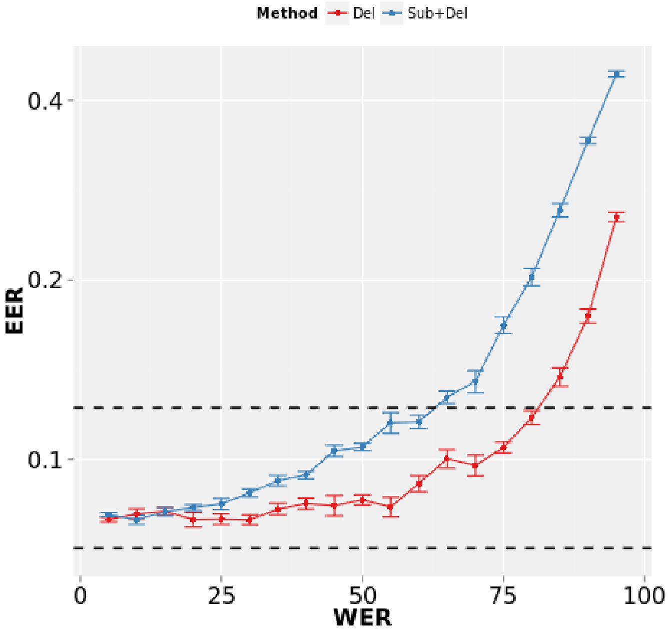


Figure 3.6: Simulated WER and topic detection EER for Fisher Spanish

an actual ASR system, substitutions are often 'sounds-like' errors, but this distinction is lost when computing WER. For comparison we also simulated systems over the same WER range but entirely with deletions.

Figure 3.6 compares topic classification performance on the simulated errorful transcripts at 5% WER intervals. We ran 10 trials for each type of error induction method at each WER point, and the standard error over the trials are indicated by the error bars. Figure 3.6 is plotted with the EER on a log scale for legibility. For the deletion-only systems, we do not see changes in classification EER larger than a single standard deviation until WER exceeds 50%. For the mixed error simulation (Sub+Del) significant changes are observed at lower WER (higher accuracy) systems.

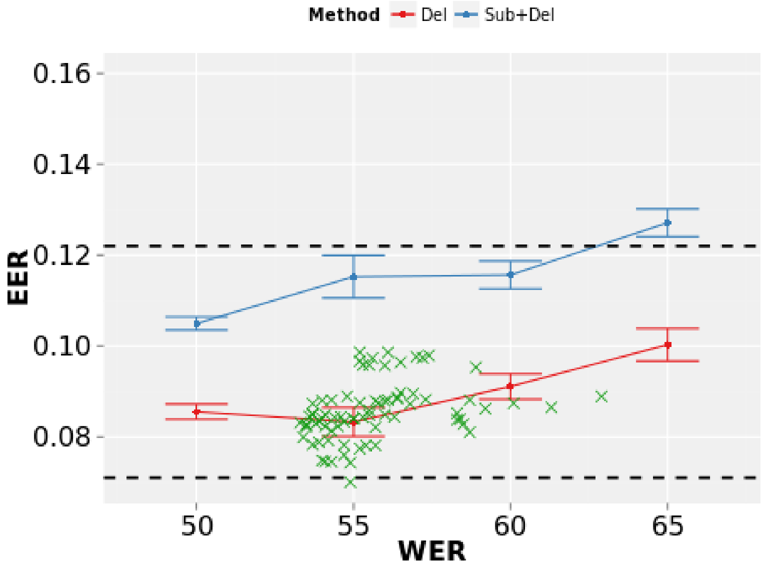


Figure 3.7: Simulated WER and topic detection EER for Fisher Spanish. Actual ASR systems in green.

The difference between the two simulations makes intuitive sense in terms of the classification task. Deletion errors suggest missing but not misleading evidence for the conversation topic. If we place the actual ASR systems on the same graph with the simulations (cf. Figure 3.7) we observe that despite exhibiting a 2 to 1 substitution to deletion ratio, the true ASR errors induce topic classification performance closer to the deletion-only simulation. For example, in the a system with 65% WER, approximately 2500 out of 291000 substitution errors differed only by the addition or subtraction of the plural ‘s’ at the end of the word - e.g. *matrimonio* vs. *matrimonios*. A random substitution is much more likely to result in a topically unrelated word.

3.2.2 Keyword Spotting Models

To present an additional perspective on where topic content begins to be lost due to speech recognition errors we also built a limited vocabulary keyword spotting system for the top 1000 keywords. In most cases the keyword spotting system does *not* permit classification performance on par with the human transcript baseline. By evaluating the higher error system in terms of keyword retrieval metrics, rather than aggregate WER, we are better able to identify which errors from the ASR system on which to focus subsequent efforts for improvement. In our analysis, ranked retrieval metrics, particularly the area under the keyword recall-precision curve (AUC), are better predictors of the ability to perform topic classification.

We construct a keyword spotting system from standard HMM-based ASR tools, again with the Kaldi framework. Our training corpus is the same as the full vocabulary system, however we assume that only instances of the top 1000 keywords are annotated for acoustic training. The remaining speech is mapped to a filler word. The language model, if useful at all, models the transition from filler word to the keywords and vice versa. However, the best retrieval performance was observed with a minimal language model scaling factor, indicating the acoustic model contained most of the useful information.

The keyword spotting follow the same training procedure as for the full vocabulary system so we obtain results for GMM, SGMM, and DNN output density models on HMM states (i.e. context-dependent triphones). Further details of the keyword spotting architecture can be found in [3].

3.2.3 Results

In contrast to the full-vocabulary systems, the keyword spotting models do not exhibit Topic ID performance on par with the transcript baseline, *except for the DNN-based models* (cf. Table 3.3). Rather than be disappointed by these results, we use this opportunity to look at the retrieval results and identify causes for the degradation.

The most noticeable difference between the DNN and GMM models is the increase in recall. On average, the DNN models recalled nearly **twice as many** keyword instances as the other models. By contrast, the GMM keyword spotters had the highest precision on the search task but the lowest overall topic performance. We may conclude that a higher false alarm rate (low precision) does not by itself inhibit topic classification performance.

Based on the recall and precision of the keyword spotters, (top portion of Table 3.3) it is tempting to argue that recall by itself is sufficient for reasonable Topic ID performance. However, the ranked retrieval performance reveals something more nuanced.

Figure 3.8 shows the keyword spotting retrieval results in terms of the mean search AUC (MAUC) of all keywords plotted against EER. The keyword spotting systems, even with DNN features, are at least 50% lower in terms of search accuracy than full-

Keyword Spotting Systems					
Acoustic Model	EER	Recall	Prec.	MAUC	TWV
GMM	0.39	0.084	**0.641**	0.043	**-0.004**
SGMM	0.29	0.172	0.545	0.079	-0.012
DNN	**0.12**	**0.379**	0.338	**0.154**	-0.038
Full Vocabulary ASR					
GMM	0.09	0.278	0.464	0.395	0.342
SGMM	0.08	0.318	0.482	**0.428**	**0.384**
DNN	**0.07**	0.269	0.458	0.433	0.384

Table 3.3: Naive Bayes Topic ID EER and keyword retrieval performance for various metrics. Paired t-test gives $p < 2 \times 10^{-16}$ between different acoustic model EER results.

vocab ASR performance. The DNN system, however is twice as accurate in terms of ranked retrieval than all other keyword spotters.

Ranked retrieval metrics reflect the order of results. Higher AUC implies that correct keyword detections are more likely at the top of the term detection results. As we generate counts for Topic ID from the results, detections at the top contribute more to our bag-of-words model. We conjecture that for the DNN models, retrieval is good enough, given sufficient recall of topic-relevant words, that false alarms that obscure the topic signal do not occur high up in the result list.

3.3 Conclusion

In this chapter we considered topic information from the perspective of the topic classification task. We have drawn two conclusions, first, topic information is sensitive

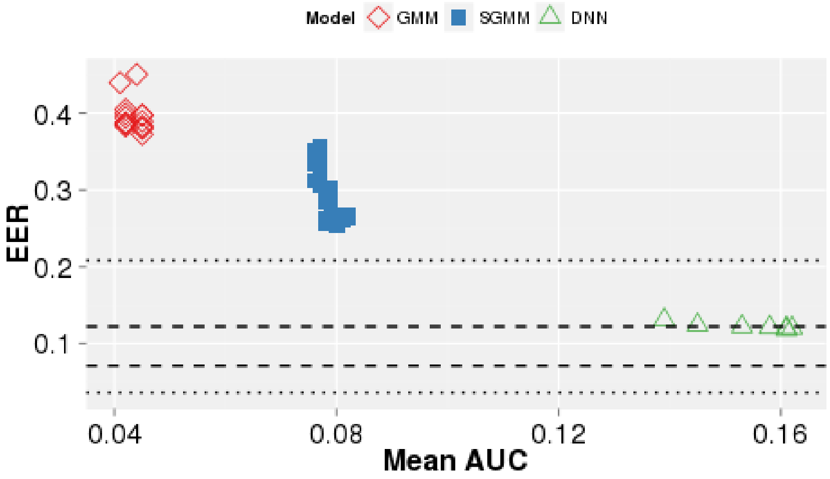


Figure 3.8: Effect of keyword accuracy on Topic EER. Dashed lines indicate Naive Bayes and SVM performance on human transcipts. Dotted lines indicate standard deviation across topics on human transcript performance.

to location, that is, a bag-of-words model that ignores location does not accurately model topic information spoken documents (with respect to classification). Secondly, we have examined the robustness of topic information in speech with respect to recognition errors. We have argued that improving keyword retrieval is more relevant to improving topic classification than is optimizing word error rate. Thus in subsequent chapters, we will be more interested in developing techniques that improve KWS performance on speech, not just WER.

Chapter 4

Topic Information and Speech Retrieval

Having considered the strength of the topic signal from the perspective of classification, we analyze the impact of topic information on the keyword retrieval task. We shift our focus from a supervised setting, where the user information need is expressed explicitly by labeled examples, to an unsupervised setting, where queries are expressed as key words or phrases. In this chapter we demonstrate how both *local* context, in terms of repetition and cache-based language models, and *broad* context, expressed as latent topic mixture models, individually and jointly, improve keyword retrieval performance. The results in this chapter motivate the joint model of both contexts that follows in Chapter 5.

To understand how word context can affect keyword performance, we describe standard approaches for generating keyword scores from ASR systems. We are going to incorporate topical word contexts primarily by modifying the ASR system's language model. Both interpolation-based approaches to adapting standard backoff N-gram language models and more recent discriminative language model re-scoring can be expressed expressed in terms of finite state transducer (FST) operations. Through the FST formulations we present a contrastive view of the adaptation approaches before looking at the empirical performance.

We then present a keyword-specific method for leveraging local (within-document) repetition in any speech retrieval system. By using only the KWS score, and treating the retrieval system as a black box, we can incorporate local context without modifying the underlying speech recognition models. This approach, which we describe in Section 4.3, is generally applicable to any KWS system and improves retrieval performance across a spectrum of language conditions. However, this approach only incorporates the repetition aspect of topic context.

By contrast, in the latter part of this chapter, we show how we can apply both latent topics (broad context) and cached N-grams (local context) directly to the recognizer's language model. By interpolating unigrams from broad topic context derived from a standard LDA topic model [43, 66] we improve retrieval performance by up to 1% absolute via lattice re-scoring (applying a new N-gram language model to the recognition output). Re-decoding with the same topic-adapted language model improves accuracy by up to 2.1%. Adding local context via cached N-grams improves performance by up to 1.6%. A cascaded approach - re-decoding with latent topics then re-scoring with cached N-grams - gives an overall absolute improvement of up to 2.4%. For all languages we consider, combining broad and local topic information into the language model outperforms each individual method.

4.1 Lattice-based Keyword Scores

In the previous chapter we briefly mentioned that when using ASR word outputs either for keyword search or classification, we typically use the posterior probability of a particular word output in computing expected counts for feature generation or for ranking keyword results. In this section define how these scores are computed and formally illustrate how the ASR/KWS output is impacted by language model-based techniques for incorporating topic information.

We start with the definition of the posterior probability of a word hypothesis w_i given the word lattice output $\mathcal{L}$ of an ASR system for a single segment of acoustic input. The posterior and subsequently the ASR/KWS scores are composed using the various ASR model likelihoods for the word w_i given the acoustic input: acoustic model l_{AM}, language model l_{LM} and HMM transition model, l_H. At this level modifications or adaptions of the language model occur.

A speech recognition lattice $\mathcal{L}$ is a directed acyclic graph representing the ASR system's hypothesized set of word sequences $W = \{w_1, ..., w_n\}$ for a fixed amount of acoustic input. The nodes (states or vertices) of $\mathcal{L}$ correspond to particular locations in the input. The arcs (or edges) are labeled with the words w_i to be output for the corresponding input. The model likelihoods for an arc a with output label w_i are captured in a weight or set of weights on a. A subset of the nodes are marked initial nodes and a subset are denoted as final. We denote a sequence of arcs staring at an initial state and ending with a final state as a path π. We denote a path that passes through the arc a with word hypothesis w_0 as $\pi[a]$. With this instantiation we can safely treat $\mathcal{L}$ as a finite state transducer (cf. [78]).

The path likelihoods $l(\pi)$ are given by multiplying the acoustic, language, and transition model probabilities along the arcs in the path (Eqn. 4.1). We compute the posterior probability of a word hypothesis $p(w_0|a, \mathcal{L})$ as the fraction of the fraction of the total likelihood captured by the paths that contain the arc a (Eqn. 4.2).

$$l(\pi) = \prod_{a_i \in \pi} l_{AM}(a_i) \cdot l_{LM}(a_i) \cdot l_H(a_i) \tag{4.1}$$

$$p(w_0|a, \mathcal{L}) = \frac{\sum_{\pi_i[a] \in \mathcal{L}} l(\pi_i[a])}{\sum_{\pi \in \mathcal{L}} l(\pi)} \tag{4.2}$$

$$S_{KWS}(h_{\mathcal{L},w_0,i}) = \sum_{a \in A(w_0)} p(w_0|a, \mathcal{L}) \tag{4.3}$$

In practice $\mathcal{L}$ often contains multiple arcs with slightly different starting times for the same word. This effectively dilutes the posterior probability for a single word occurrence across multiple arcs. In KWS systems, arcs covering similar time intervals are clustered and the arc posteriors are summed to obtain a single posterior detection score $S(w_0)$ for w_0 (cf. [79]). We distinguish the word type w_0 from the i^{th} cluster of arcs $C_i(w_0)$, which corresponds to a single KWS system output hypothesis $h_{\mathcal{L},w_0,i}$.

Our first proposed method of keyword-based repetition re-scoring (Section 4.3) operates directly on this KWS score. The rest of the methods discussed in this chapter impact the language model likelihood directly. The LM probabilities contribute to the KWS score computation through the path likelihoods, composed of individual arc likelihoods.

In many current ASR implementations, output lattices are indeed instantiated as Weighted Finite State Transducers (WFSTs). Typically the above likelihoods are captured by the weights on the WFST transitions, interpreted as *costs*, and stored as negative log-likelihoods. Standard FST operations such as determinization and minimization (cf. [80]) can be applied and are typically done in negative log space.

The output lattice $\mathcal{L}$ can also be considered a realization of the composition of an FST consisting of the acoustic input (U) for each frame (discrete input time step, typically 100ms) and the decoding graph, denoted $HCLG$, which combines the lexicon (L), language model (G), context-dependencies (C) and HMM structure (H) [78]:

$$\mathcal{L} \approx S \equiv U \circ HCLG \tag{4.4}$$

We conflate the definitions given in [78] to indicate that the actual lattice $\mathcal{L}$ is obtained by heuristically pruning the true search space S. Additionally, the lattices generated in the Kaldi WFST implementation (following [78]) conflate the language model and transition model probabilities into a "graph cost". From this we can express the path cost as:

$$-\log(l(\pi)) = \sum_{a_i \in \pi} -\log(l_{AM}(a_i)) - \log(l_{LM}(a_i)) - \log(l_H(a_i)) \tag{4.5}$$

This last has practical implications for re-scoring $\mathcal{L}$ with a new language model.

4.1.1 Language Model Adaptation

When incorporating topic information into KWS systems via the language model, we want the language model probability for each arc in Equation 4.5 to be influenced by more than the preceding 2 or 3 words. We can express the topic information from either latent topic models (broad context) or local, cached context as a lower order N-gram language model and *adapt* the baseline N-gram model using the topic or cache language model.

Many approaches for combining two N-gram language models exist (cf. [81]), however we will focus on two related techniques - *linear interpolation of probabilities* and *count merging* - and contrast them with a more recent alternative cache-based, discriminative approach (cf. [37, 82]).

If we have two N-gram models A and B, for each of which we can compute

$P_A(w_i|h_i)$ and $P_B(w_i|h_i)$ for some word w_i with history h_i then with *linear interpolation* the adapted probability is simply:

$$P_{adapted}(w_i|h_i) = \lambda P_A(w_i|h_i) + (1-\lambda)P_B(w_i|h_i) \tag{4.6}$$

On the other hand, *count merging* necessitates we maintain the N-gram frequencies underlying $P_A(w_i|h_i)$ and $P_B(w_i|h_i)$ and the new model is obtained by computing:

$$P_{adapted}(w_i|h_i) = \frac{f_A(h_i, w_i) + \beta f_B(h_i, w_i)}{f_A(h_i) + \beta f_B(h_i)} \tag{4.7}$$

In [83], Hsu showed that these two approaches were both special cases of a more general model of linear model interpolation. In both cases, the interpolation weights λ and β are empirically determined.

Once new N-gram probabilities are obtained, the new language model can readily be expressed as an FST (for example, using the algorithm presented in [84]). We denote the original language model FST as G_{NG} and the adapted model as G_{adapt}. The new G_{adapt} can be used in any other FST operations in place of the original G_{NG}, such as in constructing the decoding graph $HCLG$ as we will see subsequently, and in lattice re-scoring.

Lattice re-scoring, in the sense of replacing the existing LM probabilities (as costs) in $\mathcal{L}$ can be expressed as two FST composition operations (in an WFST framework such as Kaldi where LM and HMM costs are expressed as a single graph cost). First, the costs of the original G_{LM} can be subtracted from the lattice arcs by composing with an FST constructed by scaling the arc weights of G_{NG} by -1. Then, the new G_{adapt} can be composed with the result to add in the LM costs of the new model.

$$\mathcal{L}_{rescored} = (\mathcal{L} \circ scale(-1, G_{NG}) \circ G_{adapted} \tag{4.8}$$

The path cost of this new lattice, by rewriting Equation 4.5, captures the fact that the lattice posteriors are now computed using the new language model (cf. Eqn. 4.9). The LM costs for the word (and history) at arc a_i reflect both the N-gram and additional topic information. Equation 4.10 shows the linear interpolation form of adaptation as an example.

$$-\log(l_{rescored}(\pi)) = \sum_{a_i \in \pi} -\log(l_{AM}(a_i)) - \log(P_{adapted}(a_i)) - \log(l_H(a_i)) \tag{4.9}$$

$$= \sum_{a_i \in \pi} -\log(l_{AM}(a_i)) - \log\left(\lambda P_{NG}(a_i) + (1-\lambda) P_{topic}(a_i)\right) - \log(l_H(a_i)) \tag{4.10}$$

To present a contrasting model to the interpolation based approach, we can consider how the discriminative, trigger-based language model adaptation of Singh and Collins (cf. [37]) is expressed in this framework. The Singh-Collins model is a cache-augmented version of the perceptron-based discriminative language model from Roark et al. (cf. [82]). In brief, the Roark model trains a perceptron whose features are N-gram counts from the ASR utterance (lattice or set of N-best discrete word sequences).

The Singh-Collins model adds *trigger features* - unigrams and bigrams from a local document context outside the current utterance to the perceptron.

One interpretation of the perceptron is as a simple linear model: a dot product of a feature vector Φ with the model weights α, both of which are vectors whose dimensionality is one more than the number of observed N-gram types (Eqn. 4.11). Both models fix the first dimension Φ_0 to be the total cost (negative log-likelihood) of the current hypothesized word sequence to be re-scored, which is precisely the path cost in Equation 4.5. The other dimensions correspond to the N-gram counts in the word sequence (and in the case of the Singh-Collins model, the binary trigger features).

In terms of applying the model, a path through the lattice and a sequence of words are equivalent. Scoring a particular path with the perceptron model produces a new path cost, which thus gives new recognition (and in our case, retrieval) outputs. To use the notation in [82], where the path cost under the discriminative model $\mathcal{D}$ we have:

$$\begin{aligned} w_{\mathcal{D}}[\pi] &= \langle \Phi(\pi), \alpha \rangle && (4.11) \\ &= \alpha_0 \cdot -\log(l(\pi)) + \sum_{i=1}^{|\alpha|} \alpha_i \cdot \Phi(\pi) && (4.12) \end{aligned}$$

We want to emphasize what Roark et al. point out, namely that by applying the model to each path in the lattice, we can interpret $\mathcal{D}$ as a WFST such that the lattice obtained by composing our original $\mathcal{L}$ with $\mathcal{D}$ contains path with the perceptron-rescored weights in previous equation (4.11).

$$\mathcal{L}_{rescored} = \alpha_0 \mathcal{L} \circ \mathcal{D} \tag{4.13}$$

Although the cost of a path in this not necessarily properly normalized, if we interpret the cost as a negative log-likelihood, we obtain the following in the lattice re-scored with $\mathcal{D}$:

$$\begin{aligned} -\log(l_{rescored}(\pi)) &= \sum_{a_i \in \pi} -\alpha_0(\log(l_{AM}(a_i) \cdot P_{NG}(a_i) \cdot l_H(a_i))) + w_{\mathcal{D}}[a_i] \qquad (4.14) \\ &= \sum_{a_i \in \pi} -\alpha_0(\log(l_{AM}(a_i) \cdot l_H(a_i)) - \alpha_0 \cdot \log(P_{NG}(a_i)) + w_{\mathcal{D}}[a_i] \\ & \qquad (4.15) \end{aligned}$$

As with the models adapted via linear interpolation, the new lattice costs still contains both the original language model information plus the new model, expressed as $w_{\mathcal{D}}$, derived from the the N-gram and trigger features of the model. In some respects, the discriminative model $\mathcal{D}$ is a dynamic scaling of the original language model.[1]

$$P_{adapted} = w_{\mathcal{D}}[a_i] \cdot P_{NG}(a_i)^{\alpha_0} \tag{4.16}$$

[1] In practice, there is already a language model scaling factor applied to the language model or (as an inverse) acoustic model component. So in practice it is $\beta \log(P_{NG}(a))$ that is being carried through the above equations.

Both G_{adapt} and $\mathcal{D}$ are dynamic in principal. The probabilities represented by G depend on the new language model derived from topic context or cached N-grams, which will change from document to document, or in the cache of a cache or trigger model, from utterance to utterance. In the Roark model, one can show that $\mathcal{D}$ is static, given a set of perceptron parameters α. However, in the trigger model of Singh and Collins, the weights of $\mathcal{D}$ also depend on the trigger features, which vary from utterance to utterance.

4.2 Corpora

We evaluate these contrastive approaches under the term detection task paradigm using a variety of languages from the IARPA Babel research program [9]. The Babel task is modeled on the 2006 NIST Spoken Term Detection evaluation [7] but focuses on more limited resource conditions. We focus specifically on the *no target audio reuse* (NTAR) condition to make our method broadly applicable. This condition states that the audio may not be reprocessed after obtaining the search keywords.

The languages of the Babel program are provided under two conditions, Full LP (Language Pack) and Limited LP. The Full LP condition for a language consists of 100 hours of transcribed audio and a pronunciation lexicon. The Limited LP condition contains only 10 hours of transcribed audio and lexicon. For all of our experiments with the Babel languages, we limit acoustic and language model training to the transcribed portions of the language packs. Each corpus also contains a 10 hour transcribed development set, for which re report recognition (WER) and retrieval (TWV) performance. Transcripts for the official Babel evaluation data have not been released at this point in time. The languages we consider in this chapter include Cantonese, Pashto, Turkish, Tagalog, Vietnamese, Zulu and Tamil.[2]

4.3 Keyword Repetition Model

We first aim to leverage word repetition as a simple form of topicality. One reason words tend to repeat within a document is that words relevant to the document's subject become more likely. In Chapter 2 we illustrated how strongly topical words like 'matrimonio' occurred in bursts within documents related to that topic (cf. Figure 2.2).

However, to exploit this burstiness for keyword search, we don't need to model the document's subject matter explicitly. The bursts will occur whether we know the subject or not. We can use this phenomenon, across a spectrum of languages, to boost low-scoring keyword hits. By applying a repetition model, these hypothesized keywords, which may have been unlikely due to acoustic or language model scores, are now detected as repetitions of other detected keywords.

[2] Language collection releases babel101-v0.4c, babel104b-v0.4bY, babel105b-v0.4, babel106-v0.2g, babel107b-v0.7, babel206b-v0.1e, and babel204b-v1.1b respectively.

$$S_{BOOST}(h_{d,w,i}) = (1-\alpha)\cdot S_{KWS}(h_{d,w,i}) + \alpha \max_{j}\left[S_{KWS}(h_{d,w,j})\right] \tag{4.17}$$

Our model is a straightforward score interpolation that boosts scores for keywords seen more than once in a document. We assume we are given results from a term detection system run over a corpus of spoken documents D using a list of keywords W. The system outputs $H_{w,d}$ hypothesized detections for each document d and keyword w. We take the top-scoring hypothesis for each w, d as evidence that w occurred at least once in d. We boost every other hypothesis for w in d by the top score, so the final score depends on the underlying ASR/KWS system as well as the presence of repetition (cf. Equation 4.17).

This notation differs slightly from the KWS score computation based on ASR lattices presented earlier (cf. Eqn. 4.3) so as to illustrate that this method is not restricted to word-lattice based KWS output. This re-scoring formula can apply to any system (or combination of systems) that can generate detection hypotheses for the keywords. In practice, our experimental results do use the latter lattice-based scoring.

We demonstrate that the method generalizes well, by applying it to all 2013 Babel languages (Cantonese, Turkish, Tagalog, Cantonese and Vietnamese). We demonstrate consistent improvements in all languages in both the Full LP and Limited

LP settings, suggesting both the utility and universality of repetition phenomena.

4.3.1 Repetition Measures

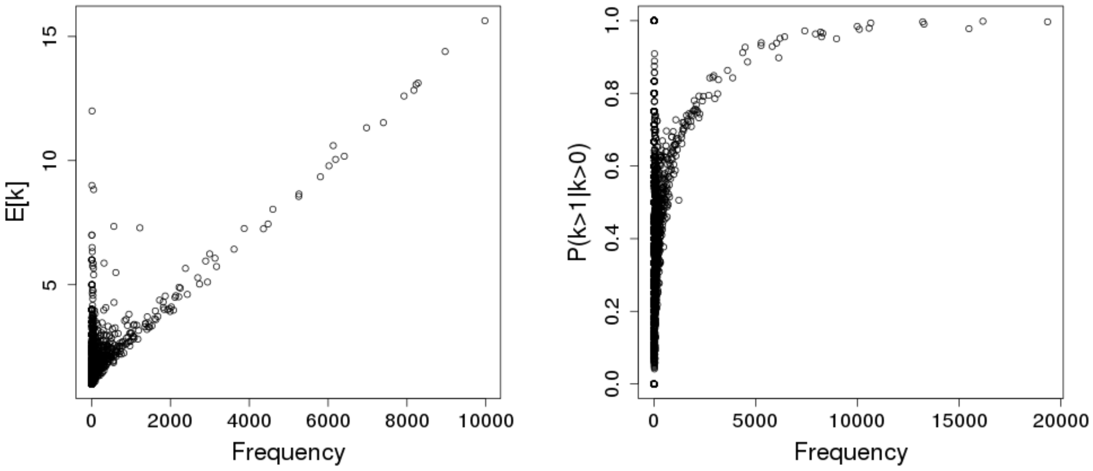
Two measures for word repetition over a corpus suggests this approach ought to be effective for a broad range of keywords: *burstiness* and *adaptation*. Burstiness (Equation 4.18) is the expected number of occurrences k of a word w per document, given that w has been seen at least once in the document. Adaptation (Equation 4.19) is the probability of a word w occurring more than once in a a document, given it is seen at least once (that is, over documents containing the word).

$$E_w[k|k > 0] = \frac{f_w}{\mathrm{DF}_w} \tag{4.18}$$

$$P_{adapt}(w) = P_w(k > 1|k > 0) \tag{4.19}$$

Figure 4.1 illustrates these two measures for each word in the Babel Tagalog 100 hour training corpus. Each point represents a word in the vocabulary, and we look at how *burstiness* and *adaptation* vary with the corpus frequency f_w of each vocabulary word w. Given a fixed size corpus, burstiness naturally increases with f_w, given the document frequency DF_w is somewhat artificially capped by the corpus size. However, in both measures, we can see a large number of low-frequency words that have significantly higher burstiness and adaptation than the general trend.

If we hold with the statement that "Low frequency words tend to be rich in



(a) Burstiness

(b) Adaptation

Figure 4.1: Burstiness and adaptation probability for Tagalog training vocabulary

content, and vice-versa," [54], then a significant number of content-rich keywords should exhibit this burstiness, and we can exploit this at search time. While we do not claim any particular threshold defines "content-rich", in the context of the Tagalog corpus, we observe that 26% of all tokens and 25% of low-frequency words ($f_w < 100$) have at least 50% adaptation. This is enough of a broad trend that we can indeed leverage this for improved search.

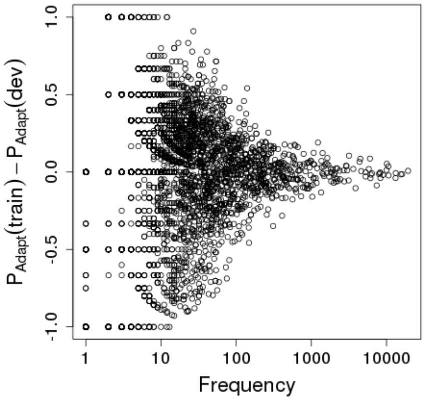
4.3.2 Interpolation

Now that we have a score interpolation model and a reasonable expectation of successful application, based on the observed strength of repetition in the data, we are faced with the choice for selecting an effective interpolation weight α (cf. Equation 4.17). How much should repetition matter? Should we vary the interpolation weight by keyword or by document?

Considering the adaptation probability, we can obtain an intuitive and effective interpolation weight directly from the training data. The actual re-scoring depends only on scores local to d, so we need only a linear pass over the results for d to obtain $\max\left[S_{kws}(h_{w,d})\right]$. Not having to re-compute α avoids incurring additional computation at search time. In addition, we showed in work published in 2014 that the effective interpolation weight also captured some inherent tendency towards repetition of each corpus. [85].

To illustrate, we consider two different methods for estimating α. First, we attempt to select different weights α_w on a per-keyword basis. Alternatively we estimate a single $\widehat{\alpha}$ for each language (from the available training data). We estimate each α based on the adaptation probability $P_{adapt}(w)$, but we find that the application is not trivial.

Using the Tagalog Full LP (100 hour) training corpus and 10 hour development set, we empirically test the two approaches. If we first compare estimates of P_{adapt} for words that occur both in the training and development sets, we find, not surprisingly, that the difference between the two estimates is only consistent for high frequency



(a) Difference in estimates of P_{adapt}

Estimate	**TWV**	P(Miss)
None	0.470	0.430
$P_{adapt}(w)$	0.423	0.474
$(1 - e^{-\mathrm{DF}_w})P_{adapt}(w)$	0.477	0.415
$\boldsymbol{\widehat{\alpha} = 0.20}$	**0.483**	**0.416**

(b) KWS Performance

Figure 4.2: Estimating interpolation weights on Tagalog Full LP corpus

words (cf. Figure 4.2a).

$$\alpha_w = (1 - e^{-\mathrm{DF}_w}) \cdot \widehat{P}_{adapt}(w) \tag{4.20}$$

$$\widehat{\alpha} = \operatorname*{Avg}_{w} [\alpha_w] \tag{4.21}$$

Given this fact, when applying adaptation values learned from the training data to the search task, we discount our per-word α estimates based on the document frequency DF_w (Equation 4.20). For a global interpolation, $\widehat{\alpha}$, we take the average over all the discounted per-word estimates. Table 4.2b shows the impact of these different choices for the interpolation weight in our keyword re-scoring formula (Equation 4.17).

The global (per-language) interpolation weight clearly outperforms any other choice in terms of keyword accuracy (TWV). The decrease in $P(Miss)$ is also an important result, because it indicates that our re-scoring does in fact boost repeated keywords above the detection threshold, increasing the number of correct keyword detections.

We include our final estimate for $\widehat{\alpha}$ with the results for the Full LP (Table 4.1) and Limited LP condition (Table 4.2). The relative values correspond to our expectations by language. The lowest values (least repetition) occur for the Turkish data, a language known for its morphological complexity, hence word units are less likely to repeat. The highest values occur for Cantonese and Vietnamese, which for the Babel program was transcribed with syllable-level word units.

4.3.3 Experiments

The complete procedure for each language in each condition (Limited or Full LP) is as follows. We first estimate adaptation probabilities from the ASR training transcripts. From these we take the weighted average as described, obtaining a single interpolation weight $\widehat{\alpha}$ for each language and training condition. We train ASR acoustic and language models from the training corpus using the Kaldi speech recognition toolkit [74] following the default Babel training recipe which is described in detail by Chen et al. [18]

Algorithm 4.1 Repetition-based term detection re-scoring

```
Estimate $\widehat{\alpha}$ on training corpus.
Train ASR Acoustic and Language Models
Decode search audio corpus.
Apply KWS algorithm
Re-score KWS results.
```

Language	$\widehat{\alpha}$	**TWV (%±)**		P(Miss) **(%±)**	
Full LP setting					
Tagalog	0.20	**0.523**	(+1.1)	0.396	(-1.9)
Cantonese	0.23	**0.418**	(+1.3)	0.458	(-1.9)
Pashto	0.19	**0.419**	(+1.1)	0.453	(-1.6)
Turkish	0.14	**0.466**	(+0.8)	0.430	(-1.3)
Vietnamese	0.30	**0.420**	(+0.7)	0.445	(-1.0)

Table 4.1: Word-repetition re-scored results for Full LP term detection corpora, improvement over baseline system denoted as percentage change.

We decode each development corpus with both Full and Limited LP models to generate ASR word lattices for the search task. We then execute Kaldi's keyword search module which is an FST-based implementation of Saraclar and Sproat's lattice-based search speech search algorithm [86]. Lastly, we re-score the search output by interpolating the top term detection score for a document with subsequent hits according to Equation 4.17 using the $\widehat{\alpha}$ estimated for the corresponding training condition. We outline these steps in Algorithm 4.1, re-iterating that our contributions, steps 1 and 5, can be carried out regardless of the ASR/KWS specifics of steps 2-4.

The overall improvements of our re-scoring algorithms are given in Table 4.1 (Full LP) and Table 4.2. In both Full LP and Limited LP settings, using only keyword repetition information, we observe improved KWS accuracy in terms of TWV between 0.7 and 1.3% absolute. Just as importantly, given TWV can be improved by either reducing false alarms or reducing misses, we decrease the miss probability in all but one condition (the exception being Vietnamese Limited LP).

The reduction in $P(Miss)$ indicates that the proposed re-scoring approach does

Language	$\widehat{\alpha}$	**TWV**	**(%±)**	P(Miss)	**(%±)**
Limited LP setting					
Tagalog	0.22	**0.228**	(+0.9)	0.692	(-1.7)
Cantonese	0.26	**0.205**	(+1.0)	0.684	(-1.3)
Pashto	0.21	**0.206**	(+0.9)	0.682	(-0.9)
Turkish	0.16	**0.202**	(+1.1)	0.700	(-0.8)
Vietnamese	0.34	**0.227**	(+1.0)	0.646	(+0.4)

Table 4.2: Word-repetition re-scored results for Limited LP term detection corpora, improvement over baseline system denoted as percentage change.

in fact do what we intend - raise the scores of repeated keywords above the system threshold. Keywords that otherwise were unlikely under either the ASR acoustic or language model are indeed boosted because they occur elsewhere in the document.

4.4 Language Model Adaptation

Whereas for the keyword repetition model we treat ASR/KWS systems as a black box, we now consider the effect of adding topic context directly to the ASR system's language model explicitly. By representing broad and local context as word N-gram probabilities that are re-computed on a document by document or utterance by utterance basis, we can use the adaptation methods described in Section 4.1 to augment the system's baseline N-gram model.

Given the augmented language model can be used either to *re-score* existing lattice output or to *re-decode* the audio to generate new lattices. We can show that adding topic context to the language model improves search accuracy in both cases, and in particular, combining both types of context (local and broad) improves accuracy of either approach individually.

Re-scoring corresponds to the re-computing the language model scores on an existing lattice $\mathcal{L}$ (cf. Eqns. 4.8,4.13). The structure of $\mathcal{L}$ and thus the words it represents are unchanged, but ideally the correct words in a more accurate model would re-score higher than incorrect words. Re-decoding, by contrast, constructs an entirely new lattice $\mathcal{L}'$, by modifying the decoding graph $HCLG$.

$$\mathcal{L}' \approx S \equiv U \circ (H \circ C \circ L \circ G_{adapted}) \tag{4.22}$$

As lattice-generation involves pruning the search space, low likelihood word hypotheses are removed from the final lattice. Changing the language model at this stage can cause a different set of words to appear in the lattice. By measuring lattice keyword recall, we can also show that by decoding with topic-augmented language models, more correct keywords survive the pruning process, which contributes to a larger search accuracy improvement.

4.4.1 Latent Topic Language Models

We represent the broad topic context of a document using a standard LDA topic model. In LDA and similar latent topic models, words and documents are modeled as arising from a document-specific mixture of $\mathcal{T}$ topics. A topic in this framework is a multinomial distribution over the corpus vocabulary - a unigram language model. A document's topic context is encoded by the inferred topic mixture for that document, $\theta^{(d)}$. We will look at parameter estimation and inference in detail in the next chapter. We obtain estimates for ϕ and $\theta^{(d)}$ then use the latter as a set of mixture weights to compute a *document-specific* unigram language model $P_T(w|\theta^{(d)})$ (cf. Equation 4.23). This document-specific model can then be used to adapted the original LM as we described previously.

$$P_T(w|\theta^{(d)}) = \sum_{t=1}^{\mathcal{T}} \theta_t^{(d)} \cdot \phi_w^{(t)} \tag{4.23}$$

For each language in our experiments we learn a latent topic (LDA) model from the training corpus transcripts. We use the Gibbs sampling approach as implemented in the Mallet toolkit [87], with minor modifications in order to allow operations on soft counts (i.e. lattice expected counts). Model estimation yields $\mathcal{T}$ topics, prior probabilities $\alpha^{(t)}$ for each topic, and the symmetric Dirichlet hyperparameter β for the unigram distributions [66]. The $\theta^{(d)}$ for the training data are computed, but not used for our task.

In order to compute $\theta^{(d)}$ for the documents in the search corpus we apply the Gibbs sampler, seeded with the learned model parameters, to expected word counts extracted from lattices generated by the baseline ASR system. Our baseline system for the experiments in this section and in Chapter 7 deep neural net (DNN) acoustic

Language	**Metric**	**N-Gram**	**LDA(R)**	**LDA(D)**
Tagalog	WER	60.8	61.6	61.6
	TWV	0.244	**0.247**	**0.254**
	L. Recall	0.778	0.778	**0.792**
Vietnamese	WER	62.0	61.9	62.1
	TWV	0.254	**0.257**	**0.269**
	L. Recall	0.555	0.555	**0.567**
Zulu	WER	67.8	68.2	68.1
	TWV	0.270	**0.278**	**0.283**
	L. Recall	0.718	0.718	**0.739**
Tamil	WER	76.0	76.1	76.2
	TWV	0.216	**0.226**	**0.237**
	L. Recall	0.573	0.573	**0.622**

Table 4.3: Recognition and retrieval performance re-scoring and re-decoding with topic-augmented LMs

models and a 3-gram backoff language model, described in detail in [49] and elsewhere. Given $\theta^{(d)}$ for the test lattices and $\phi^{(t)}$ from the training transcripts, we can compute the document-specific unigram models for adaptation. We also conducted oracle experiments using the true test transcripts to infer $\theta^{(d)}$ mixture weights on the test data, and the term detection results were identical to the fair results presented here.

We apply our topic-adapted language models to ASR/KWS systems built under the Limited LP setting in Tagalog, Vietnamese, Zulu, and Tamil. We test the topic-adapted models for both re-scoring and re-decoding the development data. We adapted the baseline model using linear interpolation (cf. Eqn. 4.6) with interpolation weight λ. We showed in [49] that we can select the λ that minimizes perplexity on the first pass one-best output, and that approach is reflected in the results in Table 4.3.

The results in Table 4.3 illustrate the importance of focusing on retrieval performance and not just word error rate (WER). The topic-adapted language models in most cases increase WER, and if that were our only metric, we would perhaps disregard the technique. However, re-scoring with topic information increases retrieval accuracy by 0.3 to 1.0% absolute and re-decoding improves keyword retrieval by 1.0 to 2.1%.

Additionally, applying topic-augmented models at decoding time increases the overall recall of keyword occurrences (Lattice Recall) from 1.2 to 4.9%. We can conclude that the topic context, which in cases where the keywords were not in the baseline lattices, can in fact boost the probabilities for topically related words such that they survive the pruning process and can be retrieved by the KWS system.

4.4.2 Cache-based Language Models

We incorporate local context by implementing a cache-augmented language model. The approach we adopt here and in [49] is based off of the work from Jelinek [34] and Kuhn [35] where we leverage the assumption that a local word or N-gram frequency estimate may be more reliable than the global frequency. Here we adapt the base language model via the *count merging* method of model interpolation. For a contrastive system, we implemented the discriminative trigger model from [37].

We define the local context for a particular utterance specifically as the expected lattice-counts for N-grams from *all other utterances in the document*. We also experimented with a exponentially decaying cache (cf. [73]) that favors adjacent utterances but found no difference in performance. For each utterance we compute an adapted language model by adding the expected lattice counts for N-grams in the local context to the original training frequencies (count-merging). In our implementation of the Singh-Collins trigger models we use the same context in computing the trigger activation.

The interpolation parameter β for the count-merging approach (cf. Eqn. 4.7) can be interpreted as a scaling factor on the local counts. We experimented with a coarse set of scaling factors, $\beta = [1, 5, 10, 20]$, but found empirically that no additional gain was found beyond $\beta = 10$. Re-estimating backoff language models on an utterance by utterance basis using the SRI Language Modeling toolkit [88] (SRILM), we required the use of the $floor$ function on the fractional expected lattice N-gram counts. This also had the effect, with $\beta = 10$ of pruning any counts with posterior probability of less than 0.1.

For the contrastive trigger model, we trained the perceptron models as described in [82], by decoding the training corpus and generating 100-best hypotheses for each training utterance. Each perceptron model on which we report was trained with 2 iterations over the data as in the previous work. For the test data we instantiated the discriminative model directly as the FST $\mathcal{D}$ and performed the composition with the output lattice.

As with the topic-specific language models, we re-score each output lattice and

Language	**Metric**	**N-gram**	**Trigger(R)**	**Cache(R)**
Tagalog	WER	60.8	61.7	**60.3**
	TWV	0.244	0.161	**0.260**
Vietnamese	WER	62.0	63.7	**61.5**
	TWV	0.254	0.190	**0.256**
Zulu	WER	67.8	69.2	**67.5**
	TWV	0.270	0.192	**0.276**
Tamil	WER	76.0	76.9	**75.5**
	TWV	0.216	0.138	**0.229**

Table 4.4: Recognition and retrieval performance re-scoring with cache-augmented LMs

apply the Kaldi lattice keyword search to the re-scored lattices. Unlike the topic-specific models, the cache-augmented language models reduce WER on the Babel development data and increases term detection accuracy from 0.2% to 1.6% absolute (cf. Table 4.4). The discriminative trigger model performs noticeably worse, the reported results arising from unigram-only trigger features, which outperformed any other feature combination from [37] when applied to the Babel data. However we would point out that the WER of the Babel systems are at least twice that of the English systems used to train the trigger models in [37] and the resulting lattice scores not well calibrated posterior probabilities, which would more negatively impact the TWV score.

Language	**Metric**	**N-gram**	**Cache(R)**	**LDA(D)**	**LDA+Cache**
Tagalog	WER	60.8	60.3	61.6	**59.7**
	TWV	0.244	0.260	0.254	**0.267**
Vietnamese	WER	62.0	**61.5**	62.1	61.8
	TWV	0.254	0.256	0.269	**0.271**
Zulu	WER	67.8	67.5	68.1	**67.4**
	TWV	0.270	0.276	0.283	**0.289**
Tamil	WER	76.0	75.5	76.2	**75.4**
	TWV	0.216	0.229	0.237	**0.240**

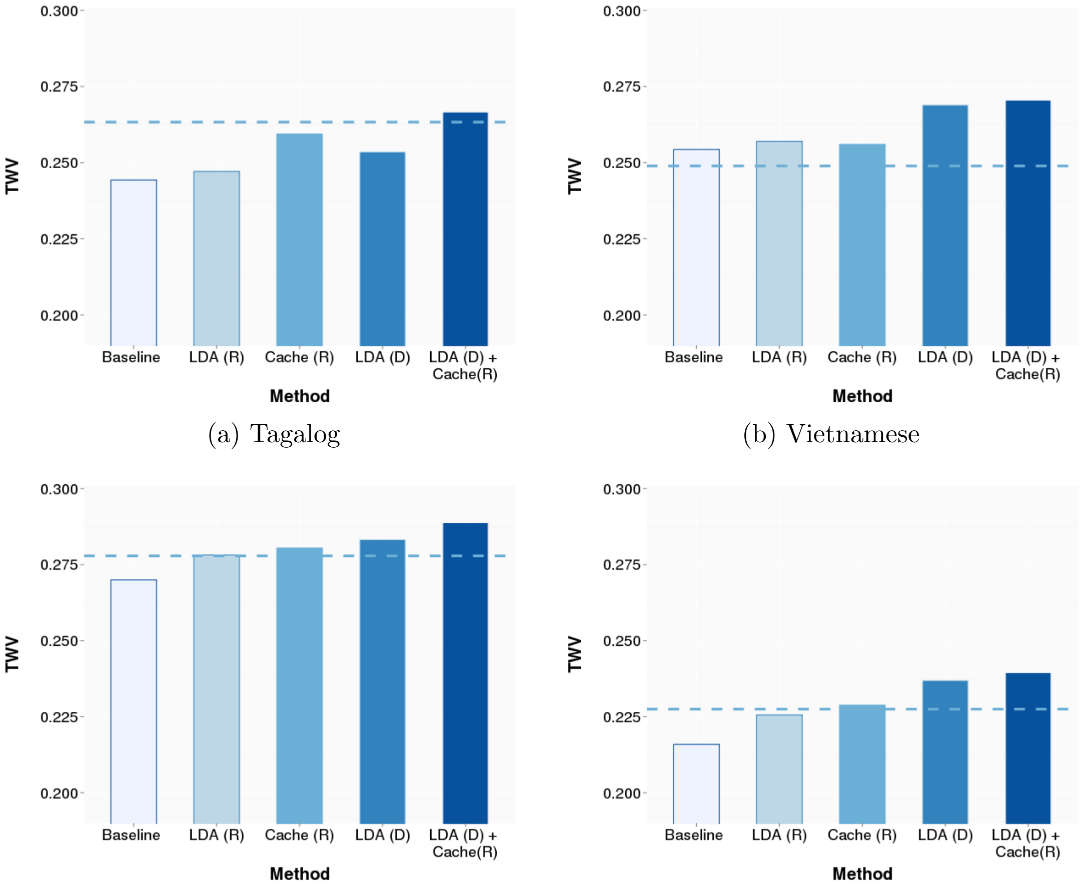
Table 4.5: Recognition and retrieval performance with topic-augmented decoding followed by cache re-scoring.

4.5 Conclusion

Lastly, we look at the broad and local contexts in terms of complementary information. In [49] we showed how we could apply the cache re-scoring after decoding with the topic-augmented language models. This result, for the Babel corpora is described for both recognition (WER) and retrieval (TWV) in Table 4.5. The cascaded result strongly suggests that the two types of topic context, while related, provide complementary information for the retrieval task.

We illustrate the overall performance impact of incorporating topic context directly into the ASR language model in Figure 4.3. The overall conclusion is as we hoped, that both broad context, implemented as topic mixture models, and local context, implemented as cached N-grams, when added to the language model, improve keyword retrieval performance.

In general, across all 4 languages re-decoding with the topic-augmented models,



(a) Tagalog

(b) Vietnamese

(c) Zulu

(d) Tamil

Figure 4.3: Term detection performance when adapting ASR language models. Dashed line indicates ad-hoc repetition performance.

denoted **LDA(D)**, outperforms simply re-scoring existing lattices, denoted **LDA(R)**. The relative performance of local context, **Cache(R)** versus the LDA models depended on the language. In Zulu and Tamil, the cache re-scoring outperformed the LDA re-scoring, but not re-decoding. In Tagalog, the cache re-scoring outperformed both **LDA(R)** and **LDA(D)**. In Vietnamese, the cached models only slightly outscored the baseline.

For comparison we include the performance of the non-LM repetition re-scoring algorithm described in the first half of the chapter, represented as the dashed line in Figure 4.3. As we might expect, this method tracks with the cache results, performing best on the Tagalog corpus, where the cache-adapted LM also out-performed other methods, and underperforms on Vietnamese, just as the **Cache(R)** approach.

The final conclusion we can draw from our experiments is that the combination of **LDA(D)** and **Cache(R)** in a simple cascade outperforms each method individually, in all 4 languages. This result leads us to assert that the two types of topic context, while related, are complementary, and in the remaining chapters we will consider a joint model of the two phenomena, with an eye towards retrieval performance.

Chapter 5

Cache-augmented Latent Topic Models

We have thus far empirically shown ways in which topic information as *local context* (repetition) and *broad context* (subject matter) can improve speech retrieval tasks. Our cascaded mixture of a standard Dirichlet-Multinomial topic mixture model (LDA) with cached N-grams suggests that jointly modeling should yield similar results. The goal of this chapter is to construct a formal model capturing both types of context and deriving the sampling distributions necessary for an efficient implementation.

Given the related space of topic and language models we aim to introduce location dependency while preserving the power-law property of Dirichlet-Multinomial distributions. Additionally, given the results of the previous chapter a LDA-style unigram topic mixture is easily interpolated with traditional backoff N-gram language models. As our goal is to apply the model in speech recognition and retrieval, we do not want to give up the proven effectiveness of short N-gram contexts.

To this end we propose a straightforward extension of the standard LDA topic model [43, 66] whereby words can be generated *either* from a latent topic or from a document-level cache. At each word position we flip a biased coin. Based on the outcome we either generate a latent topic and then the observed word, or we pick a new word directly from a document-level cache of already observed words. In this model we simultaneously learn the underlying topics and the tendency towards repetition.

The rest of this chapter is organized as follows. We first present the model in its generative form. From this we can derive the joint probability for the observed and latent variables and from the joint probability we then derive sampling distributions necessary for parameter estimation and inference via a Gibbs sampling (Markov Chain Monte Carlo) procedure.

5.1 Cache-augmented Generative Process

As with LDA, we assume documents in a corpus a generated from $\mathcal{T}$ latent topics. For this chapter, we will use the term *topic* specifically to refer to a unigram distribution over a vocabulary of size V (cf. [27]). Each topic t is denoted by $\phi^{(t)}$, a

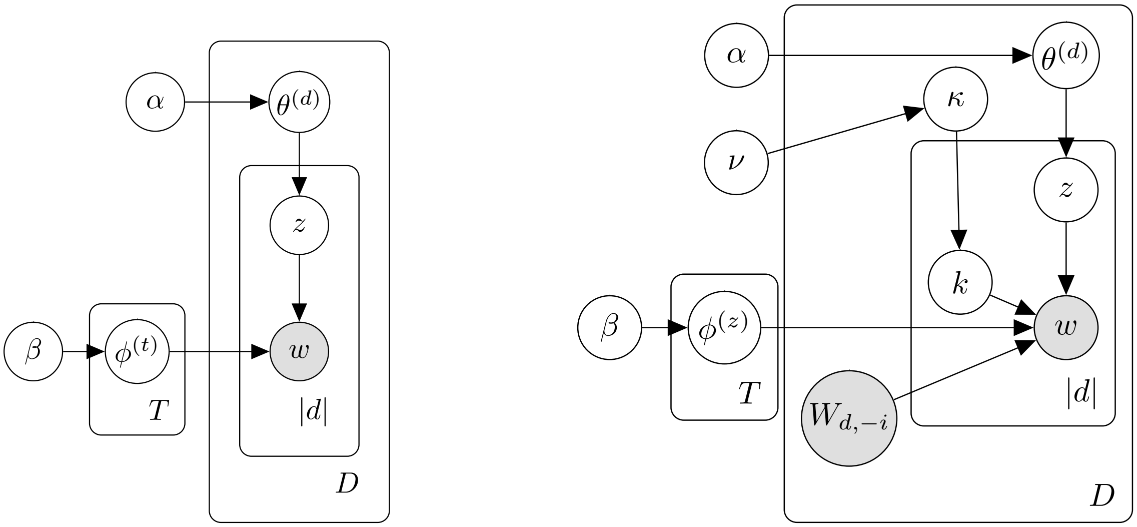
Algorithm 5.1 κ-LDA cache-augmented generative process

```
for topic $t \in \mathcal{T}$ do
    Draw $\phi^{(t)} \sim \text{Dirichlet}(\beta)$                    # Draw topic distributions
for all $d \in \mathcal{D}$ do
    Draw $\theta^{(d)} \sim \text{Dirichlet}(\alpha)$                 # Draw topic proportions
    Draw $\kappa^{(d)} \sim \text{Beta}(\nu_0, \nu_1)$                # Draw cache proportions
    for $w_{d,i}, 1 \leq i \leq |d|$ do
        Draw $k_{d,i} \sim \text{Bernoulli}(\kappa^{(d)})$            # Draw cache usage state per word
        if $k_{d,i} = 0$ then
            Draw $z_{d,i} \sim \theta^{(d)}$                          # Draw topic state per word
            Draw $w_{d,i} \sim \phi^{(t=z_{d,i})}$                    # Generate word from topic
        else
            Draw $w_{d,i} \sim \text{Cache}(W_{d,-i}, i)$             # Generate word from cache
```

Multinomial random variable with V dimensions, and the vector component $\phi_v^{(t)}$ is the probability $P(w_v|t)$. As with LDA, topics are drawn from a Dirichlet distribution with uniform prior β.

The topic mixture for a document d is also a Multinomial random variable of $\mathcal{T}$ dimensions, denoted by $\theta^{(d)}$. Each $\theta^{(d)}$ is also drawn from a Dirichlet distribution, parameterized by the $\mathcal{T}$-dimensional prior α. The vector component $\theta_t^{(d)}$ gives the probability for a topic given the document, $P(t|d)$. In terms of topic mixtures, our model acts in the same manner as an LDA model, and our implementation follows the best practices from Wallach et al. (cf. [27]) in periodically re-estimating the hyperparameters β and α.

Our primary extension of the basic LDA topic model is a formal integration of a document-level cache to the generative process and sampling mechanisms. To capture the interaction with cached local context, we introduce two additional sets of variables, $\kappa^{(d)}$ and $k_{d,i}$. The state $k_{d,i}$ is a Bernoulli random variable where a value of 1 indicates



(a) LDA

(b) Proposed Model

Figure 5.1: Plate diagrams illustrating the differences between LDA and our proposed model

that the word $w_{d,i}$ is to be drawn from the cache. A value of 0 for $k_{d,i}$ indicates that $w_{d,i}$ will be drawn from the latent topic state.

The $\kappa^{(d)}$ variable is a document specific prior on the cache state $k_{d,i}$. We intend for this latent state to capture the amount of repetition present in the document, and by extension, the corpus. We evaluate this empirically in Chapter 6. We $\kappa^{(d)}$ be a Beta prior for the Bernoulli state variables $k_{d,i}$. The Beta variable is parameterized by the terms ν_0 and ν_1, which can be equivalently expressed as a two-dimensional Dirichlet prior ν. As with the Dirichlet priors on the topic and document multinomials, the Beta-Bernoulli conjugacy allows for a straightforward formulation of the joint probability $P(W, Z, K, \Phi, \Theta, \kappa)$ for subsequent inference tasks.

This generative process is provided as pseudocode in Algorithm 5.1. For comparison we also give the generative process psuedocode for standard LDA as Algorithm 5.2. Our notation in Algorithm 5.1 for the cache distribution at a specific word $w_{d,i}$, $Cache(W_{d,-i}, i)$, is intended to convey that when $k_{d,i} = 1$, the generative distribution for $w_{d,i}$ depends only on the observed words in d. We use the shorthand $W_{d,-i}$ to denote the sequence of words in d without $w_{d,i}$, which is equivalent to the set difference: $\{w_{d,1}, \ldots, w_{d,|d|}\} \setminus w_{d,i}$

Plate diagrams contrasting our model with the standard LDA model are provided in Figure 5.1. Graphically, we can illustrate the dependence of the current word $w_{d,i}$ both on the broad topic context, via latent topic state $z_{d,i}$ as well a cache of observed words, which we denote as $W_{d,-i}$ (cf. Figure 5.1b). Within this notation, observed variables are shaded, latent variables unshaded, and the quantity at the lower right portion of the plate indicated the number of i.i.d. instances of the set of variables contained within.

We do not specify whether the cache component contains of unigrams or higher order N-grams. Neither do we need to specify the size of the cache or any decay properties given the position i in the document. Without loss of generality, we can subsequently show that the model as presented easily handles any such variations of the cache that depend only on the observed words.

Algorithm 5.2 LDA generative process

```
for topic $t \in \mathcal{T}$ do
    Draw $\phi^{(t)} \sim \text{Dirichlet}(\beta)$                # Draw topic distributions
for all $d \in \mathcal{D}$ do
    Draw $\theta^{(d)} \sim \text{Dirichlet}(\alpha)$              # Draw topic proportions
    for $w_{d,i}, 1 \leq i \leq |d|$ do
        Draw $z_{d,i} \sim \theta^{(d)}$                          # Draw word topic state
        Draw $w_{d,i} \sim \phi^{(t=z_{d,i})}$                    # Generate word from topic
```

Given this procedure and the dependencies between variables, we can factor the joint probability of observed and latent variables as follows:

$$\begin{aligned} P(W, Z, K, \Theta, \Phi|\alpha, \beta, \nu) &= P(\Phi|\beta) \cdot P(\Theta|\alpha) \cdot P(\kappa|\nu) \cdot P(W, Z, K|\Theta, \Phi, \kappa) \\ &= P(\Phi|\beta) \cdot P(\Theta|\alpha) \cdot P(\kappa|\nu) \cdot P(Z|\Theta, \Phi) \cdot P(K|\kappa) \cdot P(W|Z, K) \end{aligned} \tag{5.1}$$

We use uppercase letters to denote sequences of the variables from Algorithm 5.1 and Figure 5.1b.

Now that we can express this factorization of the joint distribution, what we are most interested in for this and other graphical models is an estimate of the posterior distribution of the latent variables given the observed data. In our cases, we want to estimate the distribution of the various $(\phi^{(t)}, \theta^{(d)}, \kappa^{(d)})$, that is the ***topics***, ***document-specific topic mixtures***, and ***document-specific cache usage***, using the observed word sequences. Without a closed-form solution, we need to turn to *approximate posterior inference*, and in particular we employ a Collapsed Gibbs Sampling approach.

5.2 Collapsed Gibbs Sampler

As was outlined in Section 2.6, in order to obtain the necessary machinery for a Gibbs sampler, we must obtain the sampling distributions for our latent variables Z and K where the value of a single latent variable is conditioned on the current state of all other variables, observed and unobserved (Equations 5.2,5.3). Here we present a derivation of both sampling distributions, how the per-document cache operates within the sampling framework, and the bookkeeping required for a reasonable implementation.

$$P(z_{d,i}|W, Z_{-i}, K) \tag{5.2}$$

$$P(k_{d,i}|W, Z, K_{-i}) \tag{5.3}$$

First, we simplify the joint probability by *collapsing* the priors - integrating over Φ, Θ, and K. This is a common technique for latent variable models such as LDA. We collapse the priors in each factor of the joint probability to transform $P(W, Z, K, \Theta.\Phi)$ from Equation 5.4 to 5.5:

$$\begin{split} P(W, Z, K, \Theta, \Phi|\alpha, \beta, \nu) = \\ P(\Phi|\beta) \cdot P(\Theta|\alpha) \cdot P(\kappa|\nu) \cdot P(Z|\Theta, \Phi) \cdot P(K|\kappa) \cdot P(W|Z, K) \end{split} \tag{5.4}$$

$$P(W, Z, K|\alpha, \beta, \nu) = P(W|Z, K, \beta) \cdot P(Z|\alpha) \cdot P(K|\nu) \tag{5.5}$$

We integrate out the priors Φ, Θ, and K in a straightforward fashion because of the Dirichlet-Multinomial and Beta-Bernoulli conjugacy and conditional independence assumptions.

The second step in the derivation is to obtain the sampling distributions from this collapsed joint distribution. It follows that:

$$P(z_{d_0,j} = t_0 | W, Z_{-j}, K, \alpha, \beta, \nu) = \frac{P(W, Z, K | \alpha, \beta, \nu)}{P(W, Z_{-j}, K | \alpha, \beta, \nu)} \tag{5.6}$$

$$P(k_{d_0,j,i} = k_0 \in \{0, 1\} | W, Z, K_{-i}, \alpha, \beta, \nu) = \frac{P(W, Z, K | \alpha, \beta, \nu)}{P(W, Z, K_{-i} | \alpha, \beta, \nu)} \tag{5.7}$$

The numerator and denominator both refer to the closed form of the collapsed joint distribution, evaluated at particular choices for W, Z, and K - *the current sampling state plus the new possible values* t_0 *and* k_0. The only difference is that the denominator does not contain terms for the state to be sampled.

We have modeled our subsequent derivations after a very useful tutorial on Gibbs sampling by Korsos and Taddy (cf. [89]). They follow the usage from Steyvers et al. where topics are represented by Φ, states by Z, per-document topic priors by Θ and hyperparameters α and β. Keeping with this notation, we now show how to obtain the collapsed form for each factor of the joint probability (Equation 5.4).

5.2.1 Notation

To make the derivations of each factor readable, we introduce a handful of count vectors and other shorthand notations. First, we use the term $I(\cdot)$ as an indicator function that returns 1 when its arguments are true and 0 otherwise. Secondly, we use the notation $B(a)$ for both the Dirichlet and Beta *pdf* normalizing factor given that the latter is a two-parameter instance of the generalized Beta function (cf. Eqn. 5.8).

$$B(a) = \frac{\prod_{i=1}^{|a|} \Gamma(a_i)}{\Gamma(\sum_{i=1}^{|a|} a_i)} \tag{5.8}$$

Count vectors that capture the current sampler state can be defined in terms of sums over indicator functions.

Each topic state variable $z_{d,i} \in Z$ is associated with a particular word $w_{d,i} = w_0$ and has an assigned state value corresponding to some topic t_0 - when the corresponding cache indicator $k_{d,i} = 0$. We aggregate the counts of states associated with a topic t_0 as vectors $C^{(t_0)}$ in the dimension of the vocabulary. Subscripted by a particular word we can express the current topic-word association as:

$$C_{w_0}^{(t_0)} = \sum_{d=1}^{D} \sum_{i=1}^{|d|} I(w_{d,i} = w_0 \wedge k_{d,i} = 0 \wedge z_{d,i} = t_0) \tag{5.9}$$

We also need to capture the current per-document topic mixture which we express by the $\mathcal{T}$-dimensional vector $N^{(d)}$, which is given by:

$$N_{t_0}^{(d_0)} = \sum_{i=1}^{|d_0|} I(k_{d_0,i} = 0 \wedge z_{d_0,i} = t_0) \tag{5.10}$$

Each cache state variable $k_{d,i} \in K$, although also associated with a particular word $w_{d,i} = w_0$, as we will see, need only contribute to a document-level count. We define two vector terms L and T to capture the number of cache versus topic states in a particular document.

$$L_d = \sum_{i=1}^{|d|} I(k_{d,i} = 1) \tag{5.11}$$

$$T_d = \sum_{i=1}^{|d|} I(k_{d,i} = 0) \tag{5.12}$$

$$\left[\sum_{d=1}^{D} T_d\right] = \sum_{t=1}^{\mathcal{T}} \sum_{v=1}^{V} C_v^{(t)} \tag{5.13}$$

As we only consider topic-word associations where the word is generated from a topic-state, the K variables divides the corpus into two parts, words generated from the topics or words generated from the cache. We necessarily have:

$$\sum_{d=1}^{D} |d| = \left[\sum_{d=1}^{D} T_d\right] + \left[\sum_{d=1}^{D} L_d\right] \tag{5.14}$$

From this, we can also describe our generative model for words by a process whereby a document author (or speaker) introduces some topical word, but then as he or she proceeds, based on a propensity for repetition ($\kappa^{(d)}$), repeats previously uttered words for either syntax or purposes of redundancy.

5.2.2 Derivations

For the term $P(W|Z, K, \beta, \nu)$ we integrate over the topic prior ϕ. The first observation we make is that the cached words do not depend at all on ϕ, given the current cache state, so we can treat the cache probabilities independent of the topic likelihoods.

$$P(W|Z, K, \beta) \tag{5.15}$$

$$= \int_{\phi} P(W|Z, K, \phi) \cdot P(\phi|\beta) d\phi \tag{5.16}$$

$$= P_C(W_{\{k_{d,i}=1\}}) \int_{\phi} \left[\prod_{t=1}^{T} \prod_{d=1}^{D} \prod_{i=1}^{|d|} P(w_i|z_{d,i} = t)^{I(k_{d,i}=0)} \right] P(\phi|\beta) \, d\phi \tag{5.17}$$

$$= P_C(W_{\{k_{d,i}=1\}}) \int_{\phi} \left[\prod_{t=1}^{T} \prod_{d=1}^{D} \prod_{i=1}^{|d|} \phi_{w_i}^{(t)\,I(k_{d,i}=0)} \right] \left[\prod_{t=1}^{T} \frac{1}{B(\beta)} \prod_{v=1}^{V} \phi_v^{(t)\,\beta_v - 1} \right] d\phi \tag{5.18}$$

$$= P_C(W_{\{k_{d,i}=1\}}) \int_{\phi} \prod_{t=1}^{T} \left[\frac{1}{B(\beta)} \prod_{v=1}^{V} \phi_v^{(t)\,\beta_v + C_v^{(t)} - 1} \right] d\phi \tag{5.19}$$

$$= P_C(W_{\{k_{d,i}=1\}}) \prod_{t=1}^{T} \left[\frac{B(\beta + C^{(t)})}{B(\beta)} \right] \tag{5.20}$$

We have denoted the cache probability of the sub-sequence of words whose cache state $k_{d,i}$ is currently set to 1 as $P_C(W_{\{k_{d,i}=1\}})$. Expanding out this term, we obtain:

$$P_C(W_{\{k_{d,i}=1\}}) = \prod_{d=1}^{D} \prod_{i=1}^{|d|} P_{cache}(w_{d,i}|W_{d,-i})^{I(k_{d,i}=1)} \tag{5.21}$$

As we stated previously, we have shown our sampler can be constructed independently of the size, order, or any other properties of the cache language model.

Moving on, to the term $P(K|\nu)$, we integrate out κ by means of the Beta-Bernoulli conjugate prior in a manner identical to the Dirichlet-Multinomial prior.

$$\begin{align} P(K|\nu) &= \int_{\kappa} P(K|\kappa) \cdot P(\kappa|\nu) d\nu \tag{5.22} \\ &= \int_{\kappa} \prod_{d=1}^{D} \left[{\kappa^{(d)}}^{L_d} (1-\kappa^{(d)})^{T_d} \right] \left[\frac{{\kappa^{(d)}}^{\nu_0} (1-\kappa^{(d)})^{\nu_1}}{B(\nu_0, \nu_1)} \right] d\kappa \tag{5.23} \\ &= \int_{\kappa} \prod_{d=1}^{D} \left[\frac{{\kappa^{(d)}}^{\nu_0+L_d} (1-\kappa^{(d)})^{\nu_1+T_d}}{B(\nu_0, \nu_1)} \right] d\kappa \tag{5.24} \\ &= \prod_{d=1}^{D} \left[\frac{B(\nu_0 + L_d, \nu_1 + T_d)}{B(\nu_0, \nu_1)} \right] \tag{5.25} \end{align}$$

The third term $P(Z|\alpha)$ can be obtained in the same manner by integrating over the topic mixtures θ. This quantity is derived in the same manner as a collapsed Gibbs sampler for standard LDA, except the topic counts must exclude words generated from cache states.

$$
\begin{aligned}
P(Z|\alpha) &= \int_\theta P(Z|\theta) \cdot P(\theta|\alpha) d\theta && (5.26) \\
&= \int_\theta \prod_{d=1}^{D} \left[\prod_{i=1}^{|d|} P(z_{d,i}|\theta^{(d)})^{I(z_{d,i}=0)} \right] P(\theta_d|\alpha) d\theta && (5.27) \\
&= \int_\theta \prod_{d=1}^{D} \left[\prod_{i=1}^{|d|} {\theta^{(d)}_{z_{d,i}}}^{I(z_{d,i}=0)} \right] \frac{1}{B(\alpha)} \prod_{t=1}^{T} {\theta_t^{(d)}}^{\alpha_t - 1} d\theta && (5.28) \\
&= \int_\theta \prod_{d=1}^{D} \left[\frac{1}{B(\alpha)} \prod_{t=1}^{T} {\theta_t^{(d)}}^{\alpha_t + N_t^{(d)} - 1} \right] d\theta && (5.29) \\
&= \prod_{d=1}^{D} \left[\frac{B(\alpha + N^{(d)})}{B(\alpha)} \right] && (5.30)
\end{aligned}
$$

Combining the closed form for the terms $P(W|Z, K, \beta)$, $P(K|\nu)$, and $P(Z|\alpha)$ we obtain the desired joint probability $P(W, Z, K|\alpha, \beta, \nu)$:

$$
P_C(W_{\{k_{d,i}=1\}}) \prod_{k=1}^{T} \left[\frac{B(\beta + C^{(k)})}{B(\beta)} \right] \prod_{d=1}^{D} \left[\frac{B(\nu_0 + L_d, \nu_1 + T_d)}{B(\nu_0, \nu_1)} \right] \prod_{d=1}^{D} \left[\frac{B(\alpha + N^{(d)})}{B(\alpha)} \right] \tag{5.31}
$$

5.2.3 Sampling Distributions

We can now use this collapsed, factored, joint probability to obtain the sampling distributions needed to update each topic state $z_{d,i}$ and cache state $k_{d,i}$. For Z we compute the sampling distribution for each possible topic value, iterating over each state, giving $\mathcal{T} \cdot |W|$ computations per iteration over the training corpus. For K, which can take on values $\{0, 1\}$, we have $2 \cdot |W|$ computations per iteration, so the overall sampling cost is still linear in the size of the corpus.

The key insight, which has been demonstrated numerous times in derivations for LDA and similar variants, is that most terms in the joint probability are unchanged when considering different values for a particular state $z_{d,i}$. Removing the particular state $z_{d_0,j}$ from the sequence only changes the count vectors $C^{(t_0)}$ and $N^{(d_0)}$ for one topic in one dimension ($v = w_{d_0,j}$). All other $B(\cdot)$ terms in other documents for other topics cancel. Also, because we are sampling a topic state, we must assume the cache variable $k_{d_0,j} = 0$. This implies that sub-sequences $W_{\{k_{d_0,i}=1\}}$ and $W_{\{k_{d_0,i}=1\},-j}$ are identical, so the cache probabilities cancel as well.

$$P(z_{d_0,j} = t_0 | W, Z_{-j}, K, \alpha, \beta, \nu) = \frac{P(W, Z, K|\alpha, \beta, \nu)}{P(W, Z_{-j}, K|\alpha, \beta, \nu)} \tag{5.32}$$

$$= \frac{P(Z|\alpha) \cdot P(W|Z, K, \beta) \cdot P(K|\nu)}{P(Z_{-j}|\alpha) \cdot P(W|Z_{-j}, K, \beta) \cdot P(K|\nu)} \tag{5.33}$$

$$= \frac{P(Z|\alpha) \cdot P(W|Z, K, \beta)}{P(Z_{-j}|\alpha) \cdot P(W_{-j}|Z_{-j}, K, \beta) \cdot P(W_j|\alpha, \beta)} \tag{5.34}$$

$$\propto \frac{P(Z|\alpha) \cdot P(W|Z, K, \beta)}{P(Z_{-j}|\alpha) \cdot P(W_{-j}|Z_{-j}, K_{-j}, \beta)} \tag{5.35}$$

$$\propto \prod_{d=1}^{D} \left[\frac{B(\alpha + N^{(d)})}{B(\alpha + N^{(d)}_{-j})} \right] \cdot \frac{P(W_{k=1})}{P(W_{-j,k=1})} \cdot \prod_{t=1}^{T} \left[\frac{B(\beta + C^{(t)})}{B(\beta + C^{(t)}_{-j})} \right] \tag{5.36}$$

$$\propto \frac{B(\alpha + N^{(d_0)})}{B(\alpha + N^{(d_0)}_{-j})} \cdot \frac{B(\beta + C^{(t_0)})}{B(\beta + C^{(t_0)}_{-j})} \tag{5.37}$$

Applying the definition of the $B(\cdot)$ function (from the Dirichlet distributions of Φ and Θ), we can further simplify the various $\Gamma(\cdot)$ expressions. For brevity, we'll express the components of vector arguments $[\alpha + N^{(d)}]_t$ and $[\beta + C^{(t)}]_v$ as a_t and c_v respectively.

$$P(z_{d_0,j} = t_0 | W, Z_{-j}, K, \alpha, \beta, \nu) \tag{5.38}$$

$$\propto \frac{B\left(\alpha + N^{(d_0)}\right) \cdot B\left(\beta + C^{(t_0)}\right)}{B\left(\alpha + N^{(d_0)}_{-i}\right) \cdot B\left(\beta + C^{(t_0)}_{-i}\right)} \tag{5.39}$$

$$\propto \frac{\prod_{t=1}^{T} \Gamma(a_t)}{\Gamma\left(\sum_{t=1}^{T} a_t\right)} \frac{\Gamma\left(-1 + \sum_{t=1}^{T} a_t\right)}{\Gamma(a_{t_0} - 1) \prod_{t \neq t_0} \Gamma(a_t)} \frac{B\left(\beta + C^{(t_0)}\right)}{B\left(\beta + C^{(t_0)}_{-i}\right)} \tag{5.40}$$

$$\propto \frac{\Gamma(a_{t_0})}{\left(\sum_{t=1}^{T} a_t\right) \Gamma(a_{t_0} - 1)} \frac{B\left(\beta + C^{(t_0)}\right)}{B\left(\beta + C^{(t_0)}_{-i}\right)} \tag{5.41}$$

$$\propto \frac{a_{t_0}}{\left(\sum_{t=1}^{T} a_t\right)} \frac{B\left(\beta + C^{(t_0)}\right)}{B\left(\beta + C^{(t_0)}_{-i}\right)} \tag{5.42}$$

$$\propto \frac{a_{t_0}}{\left(\sum_{t=1}^{T} a_t\right)} \frac{\prod_{v=1}^{V} \Gamma(c_v)}{\Gamma\left(\sum_{v=1}^{V} c_v\right)} \frac{\Gamma\left(-1 + \sum_{v=1}^{V} c_v\right)}{\Gamma(c_{w_{d_0,j}} - 1) \prod_{v \neq w_{d_0,j}} \Gamma(c_v)} \tag{5.43}$$

$$\propto \frac{a_{t_0}}{\left(\sum_{t=1}^{T} a_t\right)} \frac{\Gamma(c_{w_{d_0,j}})}{\left(\sum_{v=1}^{V} c_v\right) \Gamma(c_{w_{d_0,j}} - 1)} \tag{5.44}$$

$$\propto \frac{a_{t_0}}{\left(\sum_{t=1}^{T} a_t\right)} \frac{c_{w_{d_0,j}}}{\left(\sum_{v=1}^{V} c_v\right)} \tag{5.45}$$

$$\propto \frac{(\alpha_{t_0} + N^{(d_0)}_{t_0})}{\left(\sum_{t=1}^{T} \alpha_t + N^{(d_0)}_{t_0}\right)} \frac{(\beta + C^{(t_0)}_{w_j})}{\left(\sum_{v=1}^{V} \beta + C^{(t_0)}_{v}\right)} \tag{5.46}$$

Evaluating Equation 5.46 for each possible topic value t_0 gives a proportional set of values that can be used to sample from $P(z_{d,i})$ for each topic state. In terms of notation, this looks exactly like LDA, except the term-topic counts must not contain words in the sampler state currently drawn from the cache.

For the cache states, we can take a number of simplifying assumptions. First, if a word $w_{d_0,j}$ is going to be drawn from the cache state ($k_{d_0,j} = 1$) then the topic count vectors $C^{(t)}$ and $C^{(t)}_{-j}$, with and without state $k_{d_0,j}$ are identical, so the β terms drop out. Also, the number of topic states is unchanged (T_d) and the number of cache states differs only for document d_0. Assuming conditional of the word probabilities from the cache, as one might expect, the sample probability of the cache state depends only on the cache probability of the word in d_0 at position j (cf. Equation 5.49).

$$P(k_{d_0,j} = 1|W, Z, K_{-j}, \alpha, \beta, \nu) = \frac{P(W, Z, K|\alpha, \beta, \nu)}{P(W, Z, K_{-j}|\alpha, \beta, \nu)} \tag{5.47}$$

$$\propto \frac{P_C(W_{\{k_{d_0,i}=1\}})}{P_C(W_{-j,\{k_{d_0,i}=1\}})} \cdot \prod_{t=1}^{T} \left[\frac{B(\beta + C^{(t)})}{B(\beta + C^{(t)}_{-j})}\right] \cdot \prod_{d=1}^{D} \left[\frac{B(\nu_0 + L_d, \nu_1 + T_d)}{B(\nu_0 + (L_d)_{-j}, \nu_1 + (T_d)_{-j}}\right] \tag{5.48}$$

$$\propto P_C(w_{d_0,j}|W_{d,-j}) \cdot \frac{B(\nu_0 + L_{d_0}, \nu_1 + T_{d_0})}{B(\nu_0 + L_{d_0} - 1, \nu_1 + T_{d_0})} \tag{5.49}$$

However, if a word at (d_0, j) is to be drawn from a topic instead (Eqn. 5.50), the number of cache states (L_d) is unchanged for all documents so the cached word sequences $W_{\{k_{d,i}=1\}}$ and $W_{-j,\{k_d,i\}=1}$ are identical, and that term can be removed.

$$P(k_{d_0,j} = 0|W, Z, K_{-j}, \alpha, \beta, \nu) = \frac{P(W, Z, K|\alpha, \beta, \nu)}{P(W, Z, K_{-j}|\alpha, \beta, \nu)} \tag{5.50}$$

$$= \frac{P(Z|\alpha) \cdot P(W|Z, K, \beta) \cdot P(K|\nu)}{P(Z|\alpha) \cdot P(W|Z, K_{-j}, \beta) \cdot P(K_{-j}|\nu)} \tag{5.51}$$

$$= \frac{P(W|Z, K, \beta) \cdot P(K|\nu)}{P(W_{-j}|Z, K_{-j}, \beta) \cdot P(K_{-j}|\nu) \cdot P(W_j|\alpha, \beta)} \tag{5.52}$$

$$\propto \frac{P(W|Z, K, \beta) \cdot P(K|\nu)}{P(W_{-j}|Z, K_{-j}, \beta) \cdot P(K_{-j}|\nu)} \tag{5.53}$$

$$\propto \frac{P_C(W_{\{k_{d_0,i}=1\}})}{P_C(W_{-j,\{k_{d_0,i}=1\}})} \cdot \prod_{t=1}^{T} \left[\frac{B(\beta + C^{(t)})}{B(\beta + C^{(t)}_{-j})}\right] \cdot \prod_{d=1}^{D} \left[\frac{B(\nu_0 + L_d, \nu_1 + T_d)}{B(\nu_0 + L_d, \nu_1 + (T_d)_{-j}}\right] \tag{5.54}$$

$$\propto \frac{B(\beta + C^{(z_j)})}{B(\beta + C^{(z_j)}_{-j})} \cdot \frac{B(\nu_0 + L_{d_0}, \nu_1 + T_{d_0})}{B(\nu_0 + L_{d_0}, \nu_1 + T_{d_0} - 1)} \tag{5.55}$$

As with the sampling distribution for Z, we can expand the $B(\cdot)$ function and simplify to obtain a closed form for the K sampling distribution values. Given that the Beta distribution normalizer is a two-parameter case of the generalized $B(\cdot)$ normalizer for the Dirichlet, we get the same simplification result as in Equation 5.38.

$$P(k_{d_0,j} = 1|W, Z, K_{-j}, \alpha, \beta, \nu) \tag{5.56}$$

$$\propto P_C(w_{d_0,j}|W_{d,-j}) \cdot \frac{B(\nu_0 + L_{d_0}, \nu_1 + T_{d_0})}{B(\nu_0 + L_{d_0} - 1, \nu_1 + T_{d_0})} \tag{5.57}$$

$$\propto P_C(w_{d_0,j}|W_{d,-j}) \cdot \frac{\nu_0 + L_{d_0}}{(\nu_0 + \nu_1 + |d_0|)} \tag{5.58}$$

$$\begin{aligned}
&P(k_{d_0,j} = 0 | W, Z, K_{-j}, \alpha, \beta, \nu) && (5.59)\\
&\quad \propto \frac{B(\beta + C^{(z_j)})}{B(\beta + C^{(z_j)}_{-j})} \cdot \frac{B(\nu_0 + L_{d_0}, \nu_1 + T_{d_0})}{B(\nu_0 + L_{d_0}, \nu_1 + T_{d_0} - 1)} && (5.60)\\
&\quad \propto \frac{(\beta + C^{(t_0)}_{w_j})}{\left(\sum_{v=1}^{V} \beta + C^{(t_0)}_{v}\right)} \frac{\nu_1 + T_{d_0}}{(\nu_0 + \nu_1 + |d_0|)} && (5.61)\\
&\quad \propto \frac{\nu_1 + T_{d_0}}{(\nu_0 + \nu_1 + |d_0|)} && (5.62)
\end{aligned}$$

As the sampler depends on the prior state of Z, which is captured in the count vectors $C^{(t)}$. If the previous state of $k_{d_0,j}$ were 0, then z_j will have some topic state t_0, so the probability mass is proportional to Equation 5.61. However, if the previous state of $k_{d_0,j}$ were 1, a cache state, then we would argue that the count vectors $C^{\cdot}$ and $C^{\cdot}_{-j}$ are identical, so mass can be simplified to Equation 5.62. Other assumptions here for implementation are certainly possible. A pseudocode example for the sampler is provided in Appendix B.

5.2.4 Summary

In brief, we've derived the quantities necessary to estimate all the parameters of our proposed model using a Gibbs Sampling procedure.

$$P(z_{d_0,j} = t_0 | W, Z_{-j}, K, \alpha, \beta, \nu) \propto \frac{(\alpha_{t_0} + N_{t_0}^{(d_0)})}{\left(\sum_{t=1}^{T} \alpha_t + N_{t_0}^{(d_0)}\right)} \frac{(\beta + C_{w_j}^{(t_0)})}{\left(\sum_{v=1}^{V} \beta + C_v^{(t_0)}\right)} \tag{5.63}$$

$$P(k_{d_0,j} = 1 | W, Z, K_{-j}, \alpha, \beta, \nu) \propto P_C(w_{d_0,j} | W_{d,-j}) \cdot \frac{\nu_0 + L_{d_0}}{(\nu_0 + \nu_1 + |d_0|)} \tag{5.64}$$

$$P(k_{d_0,j} = 0 | W, Z, K_{-j}, \alpha, \beta, \nu) \propto \frac{(\beta + C_{w_j}^{(t_0)})}{\left(\sum_{v=1}^{V} \beta + C_v^{(t_0)}\right)} \frac{\nu_1 + T_{d_0}}{(\nu_0 + \nu_1 + |d_0|)} \tag{5.65}$$

At any point in the sampling procedure we can then obtain quantities for the topics, ϕ, topic mixtures θ, and cache usage κ as:

$$[\phi^{(t)}]_w = \frac{\beta_w + C_w^{(t)}}{\sum_{v=1}^{V} \beta_v + C_v^{(t)}} \tag{5.66}$$

$$[\theta^{(d)}]_{t_0} = \frac{\alpha_{t_0} + N_{t_0}^{(d)}}{\sum_{t=1}^{\mathcal{T}} \alpha_t + N_t^{(d)}} \tag{5.67}$$

$$\kappa^{(d)} = \frac{\nu_0 + L_d}{\nu_0 + \nu_1 + |d|} \tag{5.68}$$

5.3 N-gram Extension

Given this framework, it is straightforward (and has been shown elsewhere) to extend the LDA Gibbs sampling algorithm to N-grams (cf. [90]). The Topical N-gram model of Wang et al. allows for conditional formation of N-grams. An alternative approach would be to allow every word drawn from a topic distribution to also be conditioned on the preceding $(N-1)$ words.

Without any additional constraints, each topic ϕ can now be expressed as a set of multinomial distributions, one for each possible (N-1) length word history. The unigram parameter $[\phi^{(t)}]_w$ becomes $[\phi^{(h,t)}]_w$, which captures the probability of word v, conditioned on the word history h and given topic t, $P(w|h,t)$. As ϕ arises only in the sampling distribution for topic states $z_{d,i}$, it turns out we only need a slight modification to the unigram Z sampler (Equation 5.63). We only need to recompute the sub-term $P(W|Z,K,\beta)$.

First, as before, we integrate out the ϕ terms (Eqn. 5.74). Although all $\mathcal{T} \cdot V^{N-1}$ distributions appear in $P(W|Z,K,\beta)$ when we compute the sampling distribution, only the terms for the current topic and word history remain (cf. Equation 5.75).

$$P(W|Z,K,\beta) \tag{5.69}$$

$$= \int_{\phi} P(W|Z,K,\phi) \cdot P(\phi|\beta) d\phi \tag{5.70}$$

$$= P_C(W_{\{k_{d,i}=1\}}) \int_{\phi} \left[\prod_{t=1}^{T} \prod_{d=1}^{D} \prod_{i=1}^{|d|} P(w_i|h_i, z_{d,i}=t)^{I(k_{d,i}=0)} \right] P(\phi|\beta)\, d\phi \tag{5.71}$$

$$= P_C(W_{\{k_{d,i}=1\}}) \int_{\phi} \left[\prod_{t=1}^{T} \prod_{d=1}^{D} \prod_{i=1}^{|d|} \phi_{w_i}^{(h_i,t)^{I(k_{d,i}=0)}} \right] \left[\prod_{t=1}^{T} \prod_{h=1}^{V^{N-1}} \frac{1}{B(\beta^{(h)})} \prod_{v=1}^{V} \phi_v^{(h,t)^{\beta_v^{(h)}-1}} \right] d\phi \tag{5.72}$$

$$= P_C(W_{\{k_{d,i}=1\}}) \int_{\phi} \prod_{t=1}^{T} \left[\prod_{h=1}^{V^{N-1}} \frac{1}{B(\beta^{(h)})} \prod_{v=1}^{V} \phi_v^{(h,t)^{\beta_v + C_v^{(t)} - 1}} \right] d\phi \tag{5.73}$$

$$= P_C(W_{\{k_{d,i}=1\}}) \prod_{t=1}^{T} \left[\prod_{h=1}^{V^{N-1}} \frac{B(\beta^{(h)} + C^{(h,t)})}{B(\beta^{(h)})} \right] \tag{5.74}$$

As with the topics ϕ, we also index the counts for topics and words by word histories h. The counts $C^{(t)}$ from the unigram case become $C^{(h,t)}$, where $[C^{(h,t)}]_w$ is the number of occurrences of w with history h and with topic state t. Because, as with the unigram case, during sampling these count vectors only differ by 1 at any particular word, the $P(W|Z,K,\beta)$ term of the sampling proportions can now be expressed as:

$$\begin{aligned} \frac{P(W|Z,K,\beta)}{P(W_{-j}|Z_{-j},K,\beta)} &= \prod_{h=1}^{V^{n-1}} \left[\frac{B(\beta^{(h)} + C^{(h,t)})}{B(\beta^{(h)} + C^{(h,t)}_{-1})} \right] \qquad (5.75) \\ &= \frac{B(\beta^{(h_0)} + C^{(h_0,t)})}{B(\beta^{(h_0)} + C^{(h_0,t)}_{-j})} \qquad (5.76) \\ &= \frac{\beta^{(h_0)} + C^{(h_0,t)}_{w_{d,j}}}{\sum_{h=1}^{V^{N-1}} \beta^{(h)} + C^{(h,t)}_{w_{d,j}}} \qquad (5.77) \end{aligned}$$

5.4 Conclusion

We have fully described a latent topic model framework the jointly models words as either generated from a broad context (topics) or local context (cache). We have also derived the computations necessary to perform parameter estimation, by means of approximate posterior inference, using a Gibbs sampler. Our model can accommodate any type of document-level cache model that conditions the probability of a particular word only on other observed words in the same document.

Two main questions we will address in the remaining chapters. First, how well does this model capture both topic repetition properties of the data? Secondly, returning to our motivating problem, does this model generate useful language models for speech retrieval?

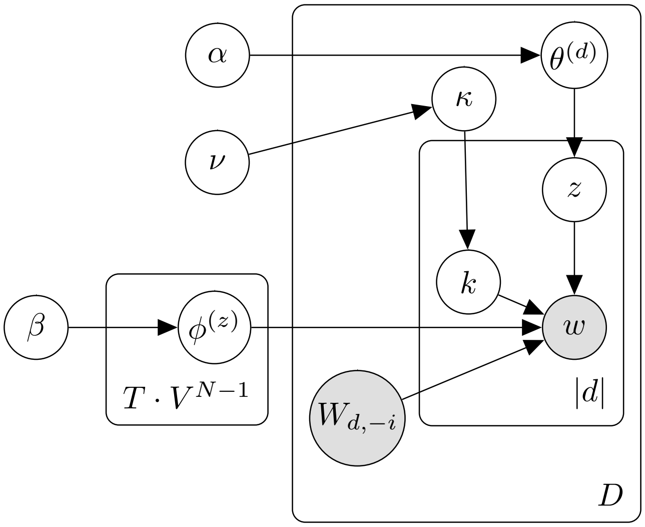


Figure 5.2: N-gram Cache-augmented Topic Model

Chapter 6

Model Analysis

Any proposed model such as the one detailed in Chapter 5 can be analyzed either *intrinsically* or *extrinsically*, with or without reference to a particular task. We evaluated standard LDA and cache-based language models extrinsically in Chapter 4 in the context of the keyword retrieval task, and we will evaluate our proposed cache-augmented topic model on the same extrinsic task in Chapter 7. In this chapter, we begin by looking directly at intrinsic, observable properties of the model, but also examine model properties through extrinsic tasks such as language modeling and topic discovery.

We estimate model parameters on informal speech corpora in a number of languages and consider the model behavior from different perspectives. We first look at the convergence and consistence of the approximate inference process itself. Given the stochastic nature of Gibbs sampling, we look at consistency and convergence across multiple iterations on the same corpora. Next we look specifically of the repetition properties inferred by our model. We ask whether the inferred cache properties correspond to our intuitions and related repetition phenomena of the data. We then look at constructing a unigram language model from the learned topic distributions and look at perplexity behavior on held out data, contrasting this with standard LDA models on the same data. Lastly, we use external topic discovery tasks to asses the quality of the 'subject matter' topic distributions.

6.1 Convergence and Consistency

In recent years, many approximate inference techniques have been well studied in the context of the standard LDA topic model, to include different implementations and optimizations (cf. [67], [91], [71]). One standard point of comparison is the *convergence* of different algorithms or models in terms of some metric. Convergence speaks to both the stability of the model and the efficiency of inference algorithm. Typically convergence can be expressed as the *likelihood* (or derived metrics, *log-likelihood* or *perplexity*) of either the training data or a held out data set under the model.

Additionally, because of the stochastic nature of Gibbs Sampling (and other MCMC) approaches we can ask how *consistent* different runs of the inference algorithm are for LDA or for our cache-augmented model. To illustrate both consistency and convergence of our proposed model, we perform 5 trials of parameter estimation on a number of similarly sized speech corpora. We can show that in terms of consistency, both LDA and our cache-augmented model are equivalently stable across trials and exhibit similar convergence behavior over time. We also analyze the convergence and consistency specifically of the $\kappa^{(d)}$ parameter under different corpora and number of topics. Again, we can show the parameter estimation converges and is stable across trials, but as intuition suggests, the behavior differs across languages.

In this chapter and in the next we focus primarily on low-resource speech recognition and retrieval scenarios. As before we utilize Limited Language Pack (Limited LP) resources from the IARPA Babel program, which contain only 10 hours of transcribed audio. The languages we consider in this chapter include Turkish, Tagalog, Vietnamese, Zulu and Tamil.[1] For interpretability of topics and cached words, we also estimate models on the CallHome Spanish corpus from LDC, which contains roughly 14 hours of transcribed conversational speech [75], LDC's Fisher Spanish transcripts [92], with 178 hours of transcribed speech, and the 359 hour subset of LDC's Fisher English transcripts we previously used for Topic ID experiments.

Corpus statistics are provided in Table 6.1. The Babel corpora are roughly all of the same size in terms of number and length (number of utterances) of documents. For speech corpora, we generally use silence-segmented utterances, roughly corresponding to a single conversation turn, instead of sentences. Sentences are generally

[1] Language collection releases babel105b-v0.4, babel106-v0.2g, babel107b-v0.7, babel206b-v0.1e, and babel204b-v1.1b respectively.

Corpus	**Docs**	**Utts/Doc**	**Tokens/Doc**	**Tokens/Utt**
Turkish	128	81.45	565.17	6.94
Tagalog	132	87.52	534.02	6.10
Vietnamese	126	80.71	932.86	11.56
Zulu	124	85.20	520.45	6.12
Tamil	125	85.77	601.57	7.01
Spanish (CallHome)	160	107.51	903.84	8.41
Spanish (Fisher)	1286	159.25	986.32	6.19
English (Fisher)	2060	189.31	1899.05	10.03

Table 6.1: Corpus sizes in terms of documents, utterances, and word tokens

not well delineated in speech transcripts. The Spanish corpora contain noticeably longer documents at least in terms of the number of utterances. English contains more words per utterance. There is some variance in terms of the number of word tokens per utterance, particularly for Vietnamese, which as has been mentioned was transcribed with syllable level word tokens.

For each corpus we analyze the training log-likelihood (per word token) over 1000 iterations of Gibbs Sampling, and averaged over 5 independent trials. We contrast the Mallet implementation of LDA with our proposed cache-augmented model (abbreviated κLDA) with either unigram or bigram cache. We also consider topic mixtures (under all models) of {50,100,150,200}.

Figure 6.1 illustrates the convergence of the per-word log-likelihood over 1000 iterations for each model condition when training on the CallHome Spanish corpus. The shaded area around the 100 topic condition indicates ± 1 sample standard deviation of the log-likelihood measurement across the 5 trials. The tightness of the

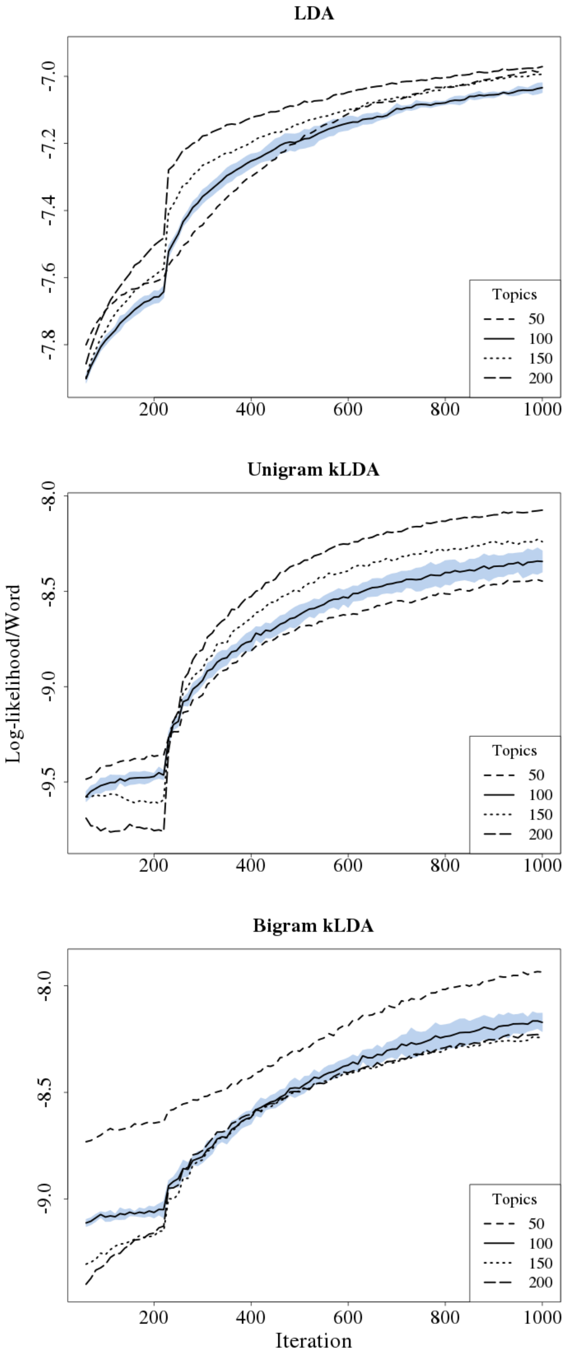
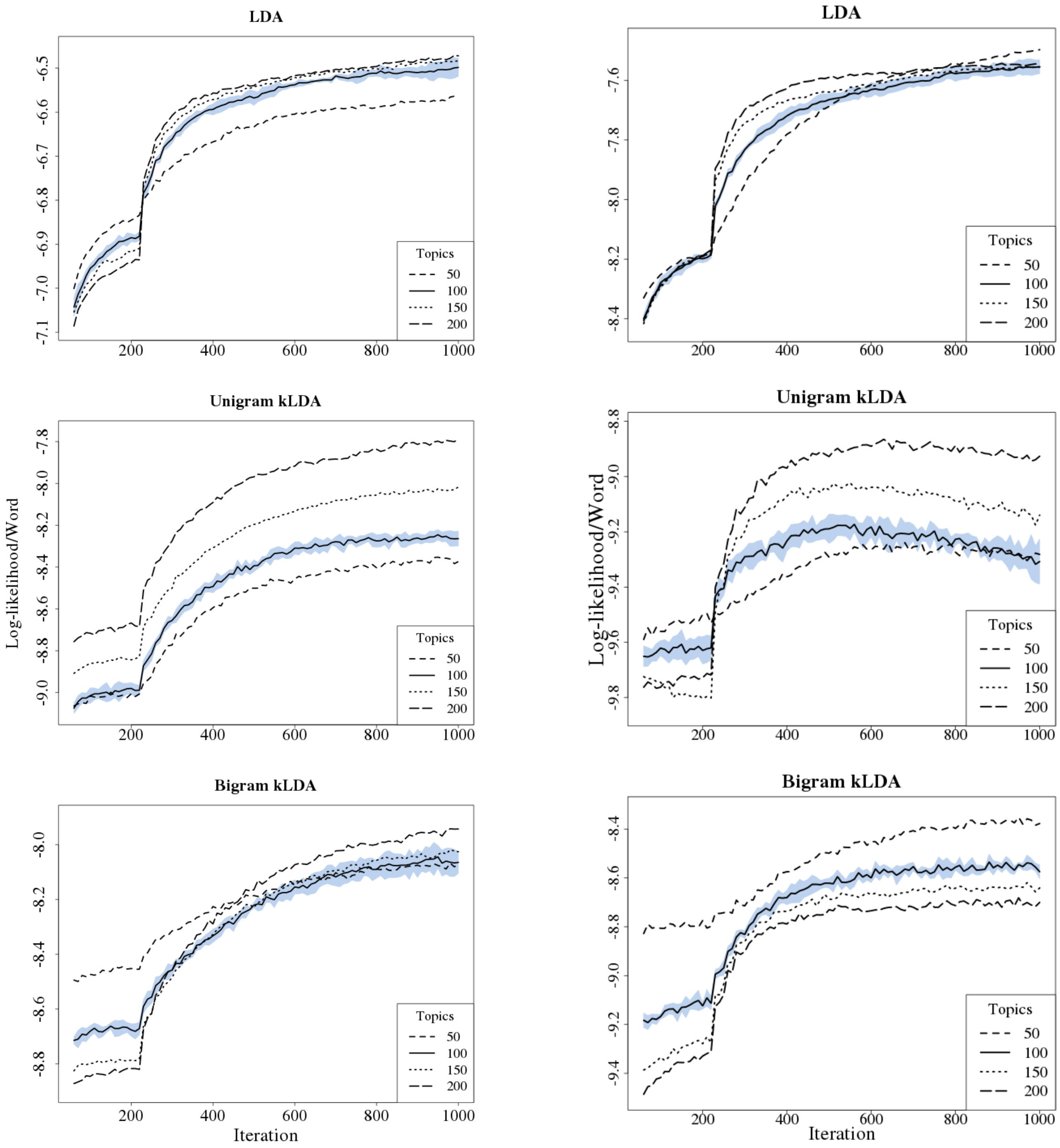


Figure 6.1: Model log-likelihood convergence over sampling process for CallHome Spanish transcripts.

log-likelihood estimates over all iterations indicates that sampling under our cache-augmented model is roughly as stable as LDA across trials.

For the sake of comparison we present the convergence figures for the Babel Vietnamese and Turkish (cf. Figures 6.2a, 6.2b). Indeed from a convergence perspective, the two behave similarly under standard LDA. However the sampling becomes significantly less smooth moving from Vietnamese to Turkish. For completeness, we include convergence figures for all corpora in Appendix C. With respect to log-likelihood convergence, in all cases the trajectory consistently changes around iteration 250, which is consistent with the application of the hyperparameter re-estimation from Wallach et al. [93] from that point on in all trials.

Alternatively, if we look at the absolute model log-likelihood we see that the cache-augmented models underperform standard LDA in both the unigram and bigram cache cases. However we will re-visit this shortly in terms of language model perplexity. Table 6.2 details the likelihood values as well as the sample standard deviation across trials (in parentheses) under all model conditions. Irrespective of the absolute value, the low variance across trials is a quantitative indication of the likelihood stability of both LDA and our proposed variants. Included are the results for the 100 topic case, with results for the 50, 150, and 200 topic case provided in full in Appendix C.



(a) Vietnamese

(b) Turkish

Figure 6.2: Model log-likelihood convergence over sampling process for Babel Vietnamese and Turkish transcripts.

Corpus	**LDA**	**κLDA-1**	**κLDA-2**
Turkish	-7.554 (0.02)	-9.173 (0.04)	-8.535 (0.02)
Tagalog	-6.523 (0.02)	-8.210 (0.03)	-7.910 (0.04)
Vietnamese	-6.498 (0.01)	-8.245 (0.03)	-8.044 (0.03)
Zulu	-7.887 (0.02)	-9.912 (0.03)	-8.594 (0.03)
Tamil	-7.887 (0.02)	-9.993 (0.04)	-8.853 (0.03)
Spanish (CallHome)	-7.034 (0.02)	-8.341 (0.04)	-8.164 (0.04)
Spanish (Fisher)	-7.505 (0.01)	-8.381 (0.01)	-8.270 (0.03)

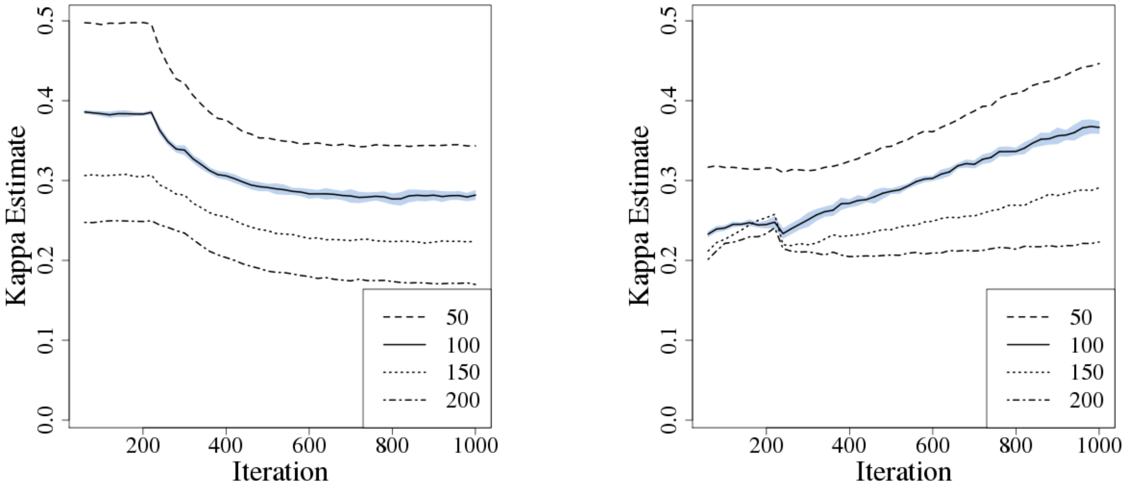
Table 6.2: Model log-likelihood per word after 1000 iterations, averaged over 5 runs, sample standard deviation in parenthesis.

6.2 Repetition

The next manner in which we can look intrinsically at the parameters output by our cache-augmented model is by analyzing to what extent the latent variables capture token *repetition* within various corpora. Within our model, repetition is captured by the cache indicator variables $k_{d,i}$ and per-document cache prior $\kappa^{(d)}$. We expect the former to be assigned to word types that tend to repeat within documents and the latter to represent the amount of repetition within a particular document, and generally this is in fact the case when looking at the data.

We continue to analyze the corpora described in the previous section, however we focus primarily on the IARPA Babel corpora, which are designed to be of equal size both in terms of length and number of documents. We will first look at estimates for the prior term $\kappa^{(d)}$ then look at individual state assignments $k_{d,i}$.

One lens through which we view how our proposed model captures repetition is the corpus κ value, defined as the mean over all documents' $\kappa^{(d)}$. We expect this value



(a) Vietnamese

(b) Turkish

Figure 6.3: Convergence and stability in κLDA sampling process for corpus κ.

to correspond to language-specific tendencies towards repetition similar to what we found in Chapter 4 with our learned interpolation weight $\widehat{\alpha}$ for repeated keywords.

As with likelihood, we also look at the convergence and stability of the estimates of κ during the sampling process. Unlike the likelihood, which we expect in general to only increase, we have no such expectation for the κ estimates. The convergence figures for Vietnamese and Turkish are shown in Figure 6.3 and the same for all languages are given in Appendix C. As with the log-likelihood across trials, the sample standard deviation for the mean of $\kappa^{(d)}$ across 5 trials was 0.01 or less for all languages and conditions, again letting us quantify the stability of the sampling procedure.

We highlight Turkish and Vietnamese as two languages whose repetition behavior we would expect to be most distinct. Morphologically the two languages are quite different. Turkish is an agglutanitive language and also exhibits vowel harmony between roots and affixes. The result of multiple affixes, plus their harmonized forms, applied to a root word results in a large number of distinct word types as compared to other corpora (see [94] for a discussion of the implications of these properties for speech recognition and language modeling). From our perspective, the addition of affixes may have the effect of turning a word token which could have been a repetition of a previous token into a new word type, lowering the likelihood of repetition.

Vietnamese, by contrast is transcribed at the syllable level and for speech recognition, N-gram language models are also applied at the syllable level, so for purposes of comparison the only available word unit is the syllable. Although it is sometimes described as 'devoid of morphology' [95] many of its units have what Noyer describes as a 'reduplicative counterpart' in which the syllable is repeated, perhaps with a change in tone to serve different syntactic or semantic roles. This, in addition to the combinatorics of a fixed alphabet and small word length limits the number of possible word types and thus increased the likelihood that any particular word type will be repeated in a particular document.

Table 6.3 lists the corpus κ values for the Babel development corpora, with Turkish and Vietnamese figures called out. Figure 6.4 shows the same estimates, and include error bars representing 1 sample standard deviation across 5 independent trials. As we would have expected, for a particular number of latent topics $\mathcal{T}$, the highest κ value is inferred from the Vietnamese corpus, and the lowest, indicating least token

Language	$\mathcal{T} = \mathbf{50}$	**100**	**150**	**200**
Tagalog	0.41	0.29	0.22	0.16
Turkish	0.31	0.19	0.13	0.09
Vietnamese	0.51	0.39	0.29	0.22
Zulu	0.33	0.26	0.21	0.16
Tamil	0.36	0.27	0.18	0.14

Table 6.3: Corpus κ inferred from 10 hour development data, by number of latent topics

repetition, is inferred from the Turkish Corpus.

We compare our κ estimates to a simple measure of repetition in each corpus, the percentage of tokens in each document that are repeated (i.e. non-singletons).

$$\textit{Document Repetition} = \frac{1}{|D|}\sum_{d \in D}\left[1 - \frac{\#\text{ types in } d}{\#\text{ tokens in } d}\right] \tag{6.1}$$

This better quantifies our intuition about the repetition within the Babel languages, as Zulu, Tamil, and Turkish have both low within-document token repetition and low corpus κ, while Vietnamese has both high κ and a high percentage of token repetition (cf. Figure 6.5).

6.2.1 Document-Level Repetition

Independent of language, a second property of the model that emerges is the overall decrease in cache usage, as captured by estimated κ as the number of latent topics increase. This is evident in Table 6.3 and Figures 6.4 and 6.5. We will consider

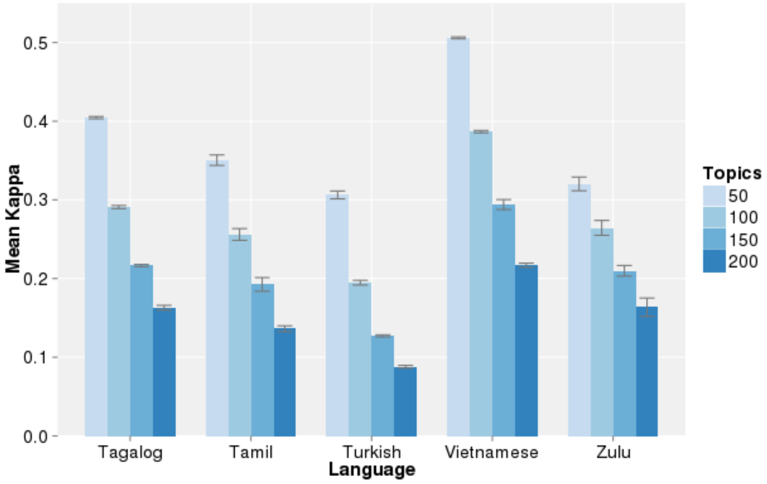


Figure 6.4: Corpus κ inferred from development corpora, averaged over 5 sampling runs.

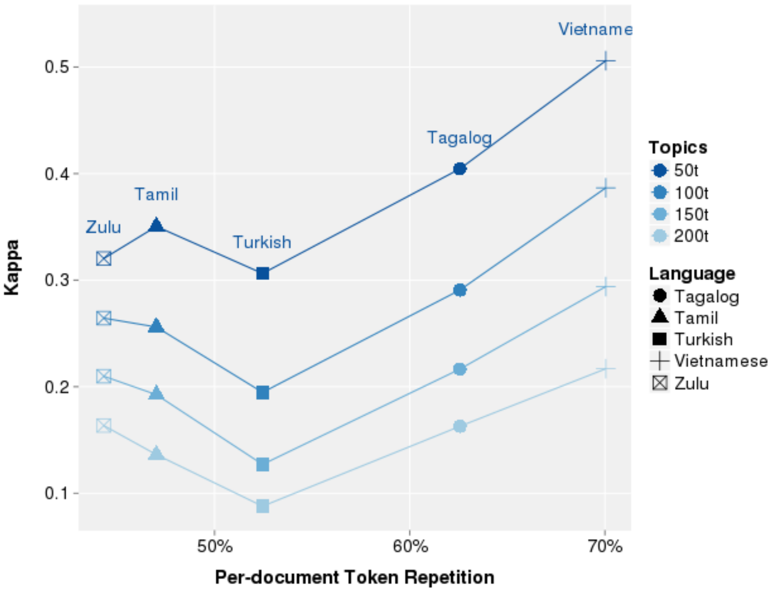


Figure 6.5: Corpus κ of development corpora, compared against the percentage of repeated tokens within each document.

Language	**$\mathcal{T}$=50**	**100**	**150**	**200**
Tagalog	0.18	0.20	0.19	0.18
Turkish	0.18	0.17	0.15	0.13
Vietnamese	0.13	0.16	0.16	0.16
Zulu	0.20	0.19	0.18	0.17
Tamil	0.15	0.16	0.15	0.14

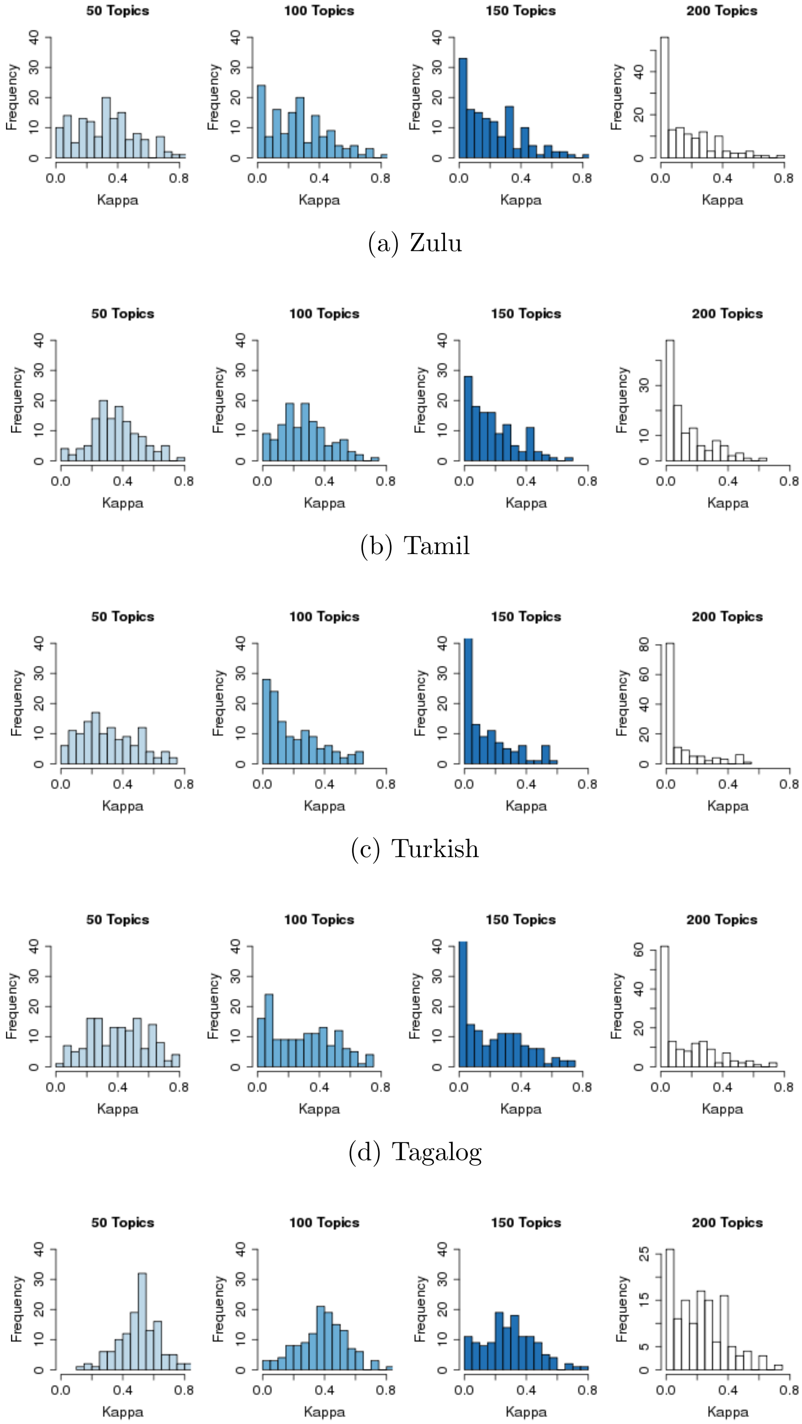
Table 6.4: Sample standard deviation of $\kappa^{(d)}$ estimates across documents from 10 hour development data by number of topics $\mathcal{T}$

within the context of retrieval in the next chapter to what extent a larger number of topics leads perhaps to overfitting and reducing the need to rely on the cache.

We can see the variation by number of latent topics in more detail by shifting our focus from the corpus level κ to the per-document estimates of $\kappa^{(d)}$. If we consider the histograms of the $\kappa^{(d)}$ estimates (cf. Figure 6.6) we can see that the variance across individual documents is not insignificant. Sample standard deviations for the data in Figure 6.6 is provided in Table 6.4.

6.2.2 Cached Word Types

To finish our analysis of the repetition patterns that are learned by our proposed model we look at the individual cache state assignments. Recall that during the generative process, each word in the corpus is assigned a latent state variable, $k_{d,i}$, indicating whether the word $w_{d,i}$ is to be sampled from the a latent topic or from the current document's cache. We consider models trained on the Fisher Spanish and English corpora in order to examine which word types are most frequently inferred



(a) Zulu

(b) Tamil

(c) Turkish

(d) Tagalog

(e) Vietnamese

Figure 6.6: Histograms of per-document $\kappa^{(d)}$ by language and number of topics $\mathcal{T}$.

as having been drawn from the cache ($k_{d,i} = 1$).

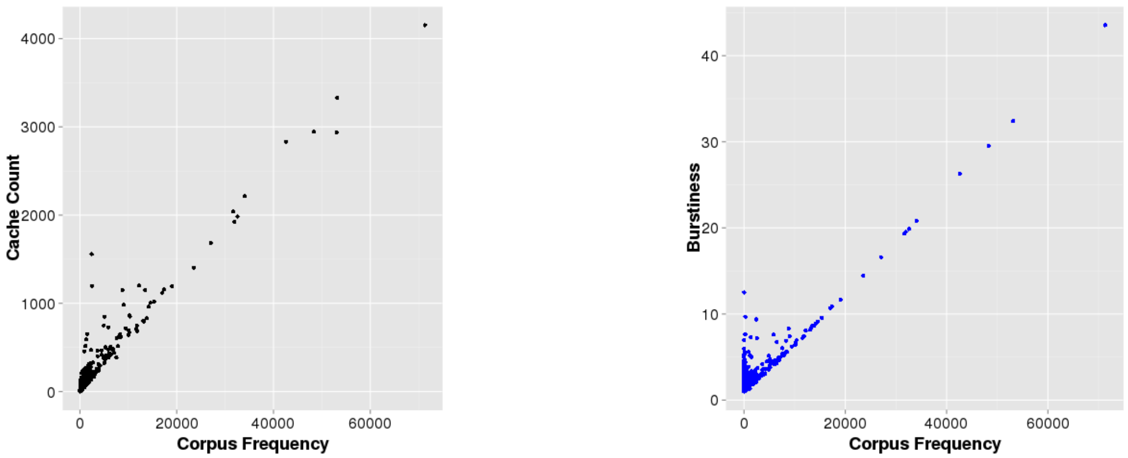
We focus on the sampler state after 1000 iterations. Each token in the training corpus is associated with a state value of $k_{d,i} = 0$ or $k_{d,i} = 1$. For each word v in the corpus vocabulary we count the number of tokens of that word type assigned a value of 1. We can define the *cache token count* (CC) over the corpus as sampling state precisely as:

$$CC(v) = \sum_{d \in D} \sum_{i=1}^{|d|} I(w_{d,i} = v \wedge k_{d,i} = 1) \tag{6.2}$$

where $I(\cdot)$ is an indicator function taking a value of 1 when its expression is true and 0 otherwise.

We can compare this quantity with other frequency measures that we considered in section 4.3.1, in particular, corpus frequency (f_w or CF), document frequency (DF), and *burstiness*, which we previously defined as f_w/DF_w. Plotting the cache token count from the final sampling state against the raw corpus frequency for each vocabulary word, we see a strong correlation, but a number of low frequency words have relatively high cache counts (cf. Figure 6.7a). This pattern also emerged when looking at word burstiness (cf. Figure 6.7b).

Again, this phenomenon appears to indicate that we are not just modeling overall frequent words with the cache states. If we take the words most frequently assigned a cache state (CC Rank) and look at how they are ranked by corpus frequency (CF Rank) and document frequency (DF Rank), irrespective of raw counts, we see that many topic words occur more frequently as cached tokens (cf. Table 6.5). The



(a) Cache counts

(b) Burstiness

Figure 6.7: Measures of repetition for the Fisher Spanish vocabulary. Each point in the graph represents a word in the training vocabulary.

highlighted topic words (chosen from the top 200 cached tokens) are clearly related to various labeled topics within the Fisher collection and occur more frequently by rank in the cache than overall in the corpus by raw or document frequency.

Although the highlighted frequently cached words are related to the labeled topics, if we compare the CC rank to the χ^2 feature selection metric (cf. [57]) only a few score highly in terms of χ^2 rank. Indeed if we look across the vocabulary (cf. Figure 6.8), we can see that χ^2 is much more strongly associated with infrequent words, both in terms of DF (Figure 6.8b) or cache sample frequency (Figure 6.8b).

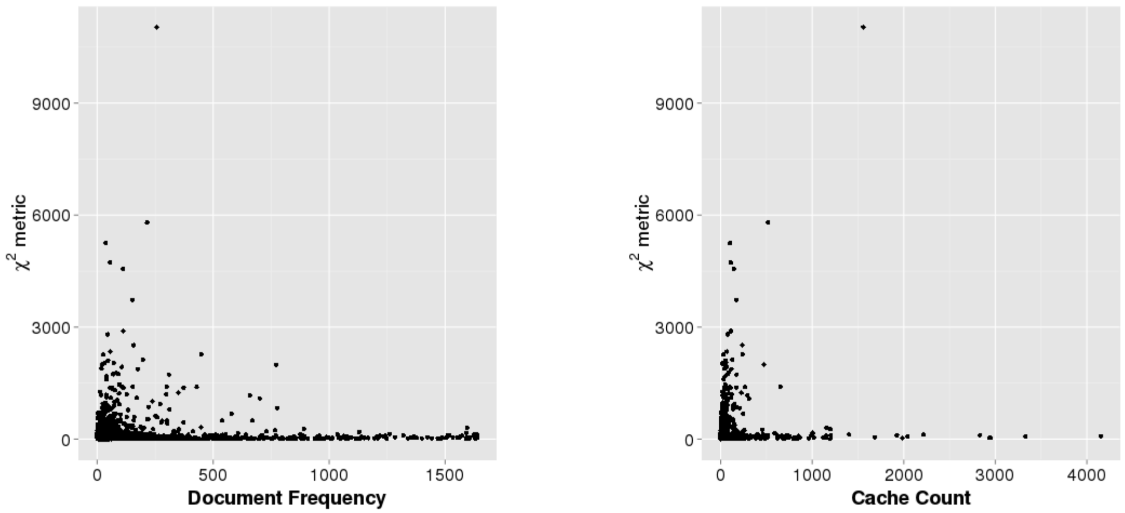
If we follow the same analysis for the Fisher English corpus we can observe the same phenomena. Our proposed cache model captures more than simply frequent words (in terms of corpus or document frequency). In Table 6.6 we again highlight the content words (indicative of the reference human topic labels) that occur within

CC Rank	Word	CF Rank	DF Rank	χ^2 Rank
1	que	1	6	1574
2	no	2	2	1821
3	de	4	7	6241
4	y	3	1	6223
5	sí	5	12	927
6	la	6	5	794
7	en	9	4	1863
8	es	7	3	8112
9	a	8	8	984
10	yo	10	10	3065
11	**música**	90	368	1
	...			
46	**religión**	177	425	2
90	**iglesia**	304	566	9
98	**teléfono**	176	277	11991
13	**york**	160	190	4493
114	**nueva**	154	174	3147
117	**dinero**	169	201	860
	...			

Table 6.5: Words in Fisher Spanish most frequently assigned a cache state of $k_{d,i} = 1$.

CC Rank	Word	CF Rank	DF Rank	χ^2 Rank
1	i	11	1	1114
2	you	5	2	913
3	and	4	3	3836
4	the	3	4	1797
5	yeah	29	8	1519
6	know	19	7	750
7	to	10	5	4512
8	a	2	6	1192
9	that	9	9	780
10	like	13	12	1696
	...			
73	**school**	213	123	59
77	**watch**	317	160	24
84	**family**	269	168	9
88	**minimum**	1018	278	2
91	**wage**	1093	292	1
93	**money**	209	129	142
95	**dog**	879	282	4
103	**computer**	506	270	21
	...			

Table 6.6: Words in Fisher English most frequently assigned a cache state of $k_{d,i} = 1$.



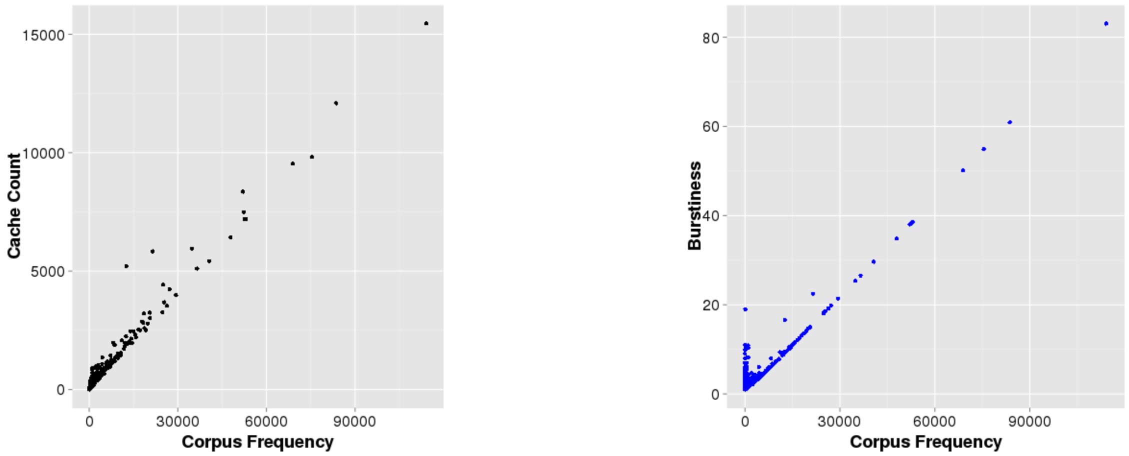
(a) Cache counts

(b) Burstiness

Figure 6.8: Measures of topic relevance and repetition for the Fisher Spanish vocabulary. Each point in the graph represents a word in the training vocabulary.

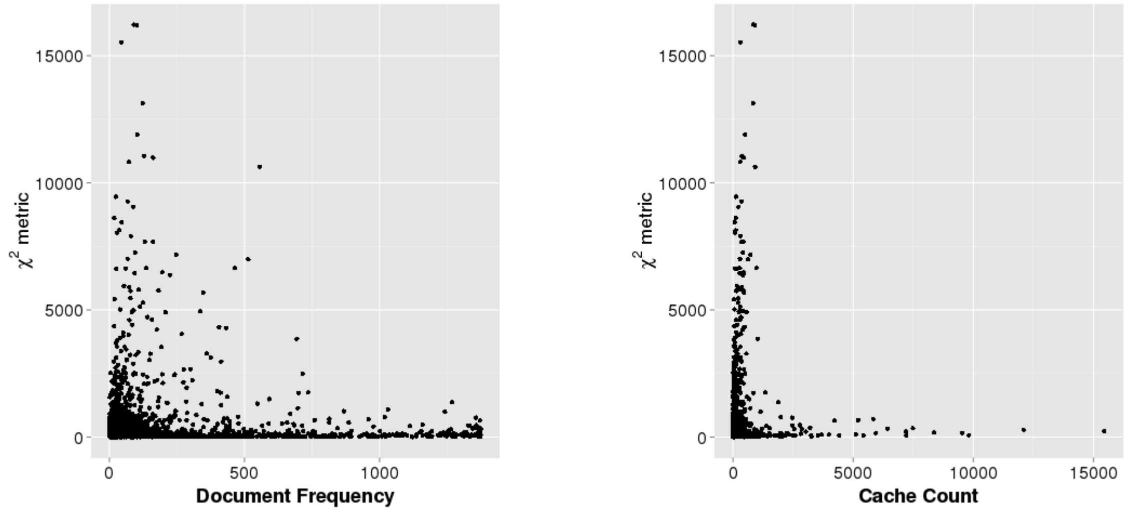
roughly the top 100 cached words. Figure 6.9 contrasts the cached state frequencies to the burstiness measure, showing trends similar to the Spanish corpus. Likewise, Figure 6.10 also shows how words that obtain a high χ^2 score relative to the reference topic labels tend to occur with low document frequency and cache usage.

Given the examples from the English and Spanish corpora, we can see that the cache state in our model captures unique properties of the given languages: not simply frequency, not capturing just an additional latent topic. In the following section we will look at our proposed model as a unigram language model and consider to what effect our modeling of repetition contributes to that task.



(a) Cache counts (b) Burstiness

Figure 6.9: Measures of repetition for the Fisher English vocabulary. Each point in the graph represents a word in the training vocabulary.



(a) Cache counts (b) Burstiness

Figure 6.10: Measures of topic relevance and repetition for the Fisher English vocabulary. Each point in the graph represents a word in the training vocabulary.

6.3 Language Models

Up to this point we have looked primarily at *intrinsic* properties of the model, directly observable or measured from the estimated parameters on our various corpora. Beginning in this section we begin to shift our focus to external tasks, starting with language modeling. As we described in the previous chapter, once we have obtained the topic proportions for a document (denoted as $\theta^{(d)}$), it is straightforward to obtain a document-specific unigram language model as a mixture of the topic distributions $\phi^{(t)}$ (cf. Eqn. 6.3).

Given the topic distributions and topic proportions we can generate these document-specific unigram LM's either from standard LDA topic models or from our proposed model. Under our model we can also incorporate the probabilities from the cache frequencies essentially on a word by word basis (cf. Eqns. 6.4,6.5).

$$P_d(w_i) = \sum_{t=1}^{T} \theta_t^{(d)} \cdot \phi_i^{(t)} \tag{6.3}$$

$$P_{cache} w_i = \frac{f_{cache}(w_i)}{\sum_{j=1}^{|V|} f_{cache}(w_j)} \tag{6.4}$$

$$P_{d+cache}(w_i) = \kappa^{(d)} P_{cache}(w_i) + (1 - \kappa^{(d)}) \cdot P_d(w_i) \tag{6.5}$$

Given these unigram language models we can look at the performance of LDA and our proposed model in terms of perplexity on the held-out data sets. Here we take the document-level cache prior $\kappa^{(d)}$ as a natural interpolation weight (Eqn. 6.5).

Language	$\mathcal{T}$	**LDA**	κ**LDA**$'$	κ**LDA**
Tagalog	50	142.90	163.30	**134.43**
	100	136.63	153.99	**132.35**
	150	139.76	146.08	**130.47**
	200	128.05	141.12	**129.94**
Vietnamese	50	257.94	283.52	**217.30**
	100	243.51	263.03	**210.05**
	150	232.60	245.75	**205.59**
	200	223.82	234.44	**204.25**
Zulu	50	**183.53**	251.52	203.56
	100	**179.44**	267.42	217.11
	150	**174.79**	269.01	223.90
	200	**175.65**	252.03	217.89
Tamil	50	**273.08**	356.40	283.82
	100	**265.02**	369.18	297.68
	150	**259.42**	361.79	301.92
	200	**236.30**	341.32	298.26
Turkish	50	**273.08**	356.40	283.82
	100	**265.02**	369.18	297.68
	150	**259.42**	361.79	301.92
	200	**236.30**	341.32	298.26

Table 6.7: Perplexity of topic-mixture unigram language models with and without unigram cache

We contrast the perplexity under three conditions on the 10 hour Babel development corpora. First, taking the $\theta^{(d)}$ and $\phi^{(t)}$ from standard LDA models, second using only the topic mixtures for our proposed model (denoted κLDA$'$), and third, our full proposed unigram model of topic mixtures interpolated with cache probabilities (denoted κLDA).

We can see from Table 6.7 that perplexity in general decreases as the number of latent topics $\mathcal{T}$ increases. However, as we will see in Chapter 7, this is not necessarily predictive of the best retrieval performance. By themselves, the topic mixtures from our proposed model underperform LDA in terms of perplexity. However, when the cache probabilities are mixed in, for Tagalog and Vietnamese the perplexity under our proposed model is relatively 3 to 15% lower than the perplexity under LDA.

6.4 Topic Discovery

While perplexity measures give us a notion of how well our proposed model explains the development data in a general sense, we would like to have some measure of how well our proposed cache-augmented model is able to extract the ‘subject matter’ of the various corpora. We wish to avoid presenting list of most frequent words in the learned topic distributions, which, though a compelling demonstration of the learning capability of topic models for English, is nonetheless still subjective in nature.

We will instead extend the analysis followed by May et al (cf. [96]) which looks at both the extrinsic performance of topic models as low-dimensional feature representations for classification, but also at the *topic discovery* task, where topic distributions are evaluated as clusterings of the data against a gold standard. What we find is that in terms of classification performance, our proposed model performs slightly worse than a typical LDA model

In order to compare against a gold standard set of topic labels, we restrict the analysis to the labeled LDC Fisher English and Spanish transcripts, with training and testings splits consistent with our previous published work (cf. [17, 32]). Classification based measures of latent topic models indirectly evaluate the learned topic distributions by posing the question, are the latent topics assigned to each document predictive of the labeled topic in terms of *effective features for classification*? Cluster-based measures ask the question, are the documents generated from the same latent topics also *assigned the same topic label* by a human annotator?

6.4.1 Classification

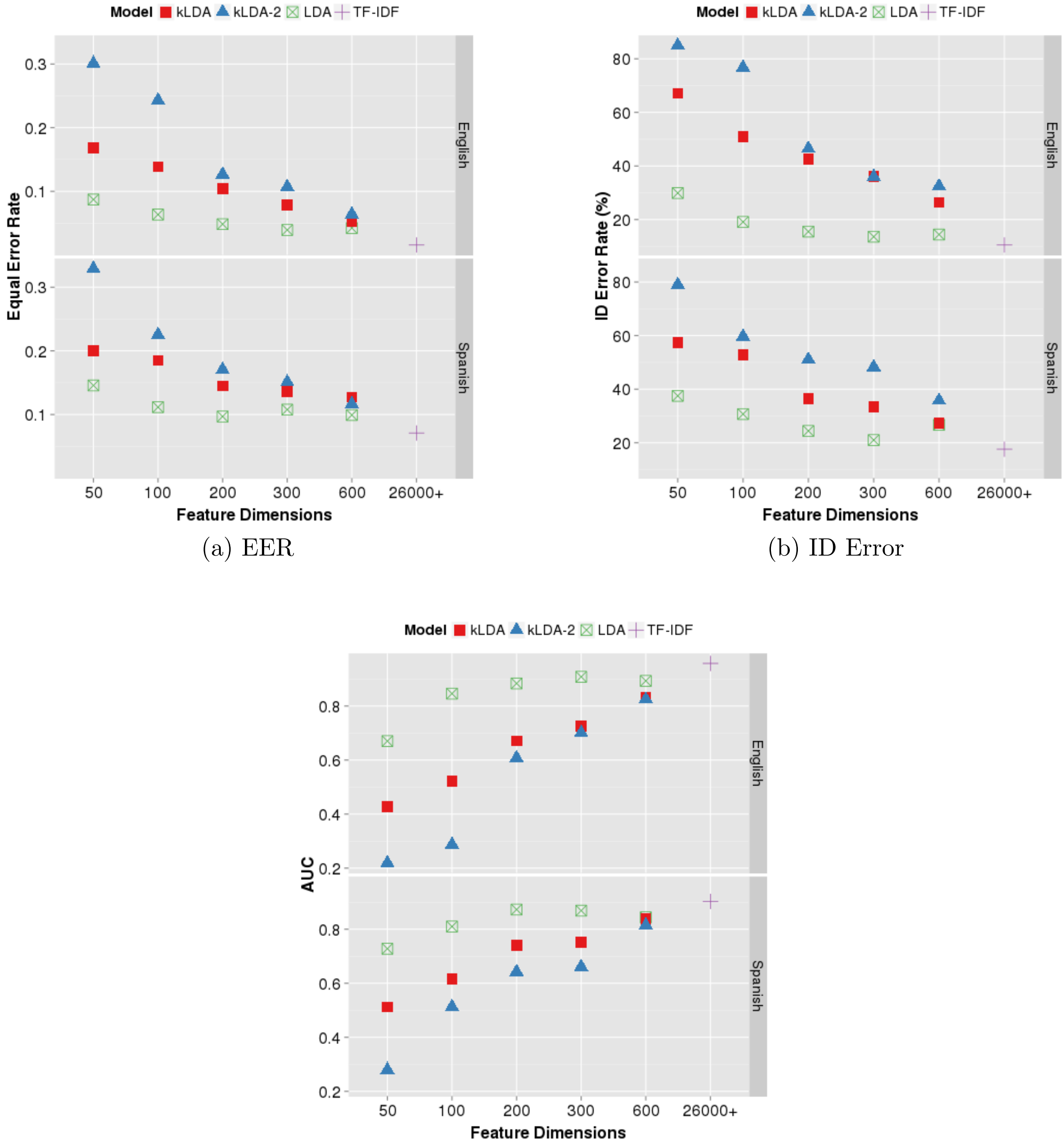
In terms of the first question, we use the topic proportions $\theta^{(d)}$ for each document as a $\mathcal{T}$-dimensional feature vector where $\mathcal{T}$ is the number of latent topics. We extract $\theta^{(d)}$ for our cache-augmented model (denoted κLDA) using the Gibbs sampling formulation detailed in Chapter 5. We also extract $\theta^{(d)}$ using the Mallet implementation of LDA. Comparison of these two models gives us an indication of what if any ability to capture the subject matter is lost when words are modeled as generated from the cache in our model.

We train topic models with $\mathcal{T} = \{50, 100, 200, 300, 600\}$ under our model and LDA, inferring $\theta^{(d)}$ for both train and test partitions of the Fisher English and Spanish transcripts. We train N 1-vs-all binary classifiers, where $N = 40$ for Fisher English, and $N = 25$ for Fisher Spanish. All results reported are averaged over all N classifiers. For a state-of-the art baseline we use TF-IDF weighted bags-of-words features using the full training partition vocabulary in each corpus (26606 and 30170 respectively). As with previous experiments, we report detection Equal Error Rate (EER), Identification Error Rate (ID Error) and Area Under the recall-precision Curve (AUC) for each system. General trends are consistent across these metrics, but reflect different application scenarios.

We capture the results of both English and Spanish classification tasks in Figure 6.11. From the perspective of classification we can conclude that our cache-augmented model, for all but the largest number of latent topics, lose some ability to capture the labeled topic signal in order to model repetition, vis-à-vis the LDA topic-only model. This is best visualized as the gap between the performance of the feature vectors $\theta^{(d)}$ inferred from LDA versus κLDA and κLDA-2 (bigram cache model).

In hindsight the results in Figure 6.11 follow naturally from the analysis of κ estimates in Section 6.2. As the number of latent topics increase the cache usage decreases, as measured by $\kappa^{(d)}$ for each document (cf. Figure 6.4). We might expect the topics learned by our cache-augmented model to approach those learned by the original LDA model as the number of latent topics grows, and from the perspective of classification, this is indeed the case.

It is worth noting again that the classification metrics are an indirect measure of how well the aforementioned models capture 'subject matter' behavior, viewed through the lense of a single set of human annotations. The results in Figure 6.11 suggest that for fewer number latent topics, the cache-augmented models differ significantly from LDA in terms of their discovered topics. However, if we consider a cluster-based evaluation, we may conclude that this difference in models (LDA versus



(a) EER

(b) ID Error

(c) AUC

Figure 6.11: Classification performance using latent topic features.

κLDA) is in part application-specific.

6.4.2 Clustering

As in May et al. [96], we also evaluate *topic discovery* of the models in terms of *V-measure* [97]. V-measure is defined as the harmonic mean of two desirable properties *homogeneity* and *completeness*, similar to F-measure, in which degenerate solutions can result in perfect recall or precision, but not both. Likewise a degenerate clustering can be obtained where all documents are assigned a single cluster ($c = 1$), or where each document is given its own cluster or latent topic ($h = 1$).

The formal definitions of homogeneity h, completeness c, and V-measure V_β follow. V-measure can be parameterized by a β, a specific preference for h versus c. All of our results report V_1 where $\beta = 1$ and homogeneity and completeness are weighted equally. The metric depends on the contingency table A whose entries a_{ck} are the number of documents assigned to class (labeled topic) c and cluster k. As in [96] we assign cluster membership based on the most likely latent topic for both LDA and κLDA.

$$V_\beta = \frac{\beta + 1 \cdot h \cdot c}{\beta \cdot h + c)} \tag{6.6}$$

$$h = \begin{cases} 0 & \text{, if } H(C) = 0 \\ 1 - \frac{H(C|K)}{H(K)} & \text{, else} \end{cases} \tag{6.7}$$

$$H(C|K) = - \sum_{k=1}^{|K|} \sum_{c=1}^{|C|} \frac{a_{ck}}{N} \log \frac{a_{ck}}{\sum_{c=1}^{|C|} a_c k} \tag{6.8}$$

$$H(C) = - \sum_{c=1}^{|C|} \frac{\sum_{k=1}^{|K|} a_{ck}}{N} \log \frac{\sum_{k=1}^{|K|} a_{ck}}{N} \tag{6.9}$$

$$c = 1 - \frac{H(K|C)}{H(K)} \tag{6.10}$$

$$H(K|C) = - \sum_{c=1}^{|C|} \sum_{k=1}^{|K|} \frac{a_{ck}}{N} \log \frac{a_{ck}}{\sum_{k=1}^{|K|} a_c k} \tag{6.11}$$

$$H(K) = - \sum_{k=1}^{|K|} \frac{\sum_{c=1}^{|C|} a_{ck}}{N} \log \frac{\sum_{c=1}^{|C|} a_{ck}}{N} \tag{6.12}$$

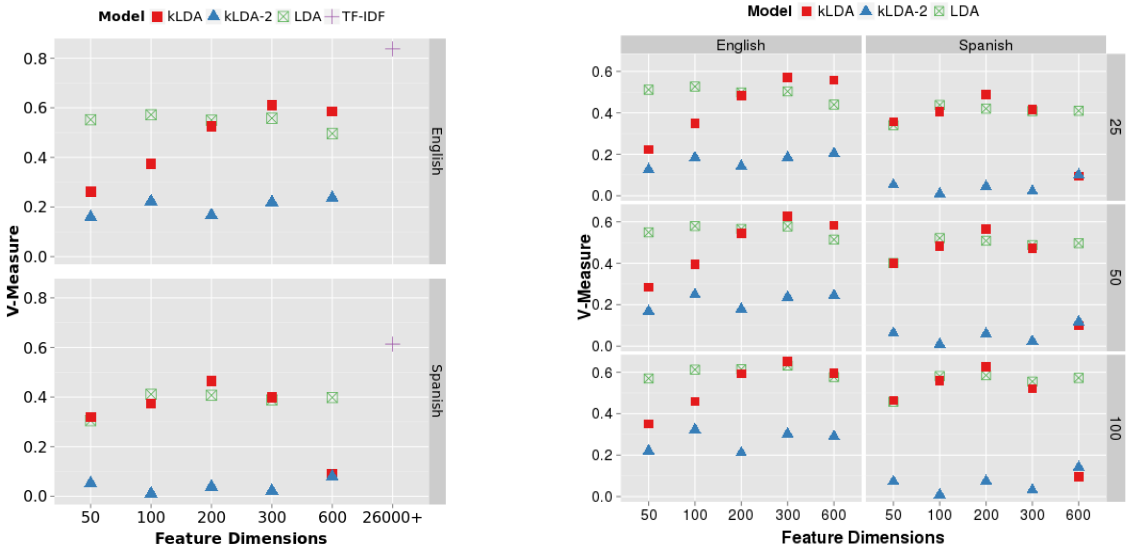
We use the same topic models from the previous section, trained with $\mathcal{T} = \{50, 100, 200, 300, 600\}$. Again we obtain the inferred topic proportions $\theta^{(d)}$ for each document d in the training partition. Taking the most likely topic t ($\operatorname{argmax}_t \theta_t^{(d)}$) as the cluster assignment for d, we compute V_1 for the topic model induced clustering and some set of class labels C over the training documents.

We consider two choices for class labels. We can use the human topic labels from the Fisher corpora as a 'gold standard' set of classes C for both English and Spanish (where $|C|$ is 40 and 25, respectively). However we can also take an unsupervised clustering of the transcript bags-of-words as an alternate set of classes. For the latter comparison, we compute clusters on the training data for each corpus of sizes $|C| = \{25, 50, 100\}$. The latter approach is a viable measure when we have no ground truth topic labels. All of the bags-of-words clustering against which we compare were generated from the training transcripts using the CLUTO toolkit's [98] *vcluster* tool with default settings.

The cluster analysis of the LDA and κLDA topic distributions gives a different perspective from the classification task to the question, how well do the models capture topic content in the corpora considered? Whereas in the classification task, there was a consistent gap between LDA and κLDA performance the cluster accuracy with respect to the human class labels (cf. Figure 6.12a) is affected by the addition of the cache model for some models but not for all. Indeed for most of the Spanish models, the V_1 performance is similar. In absolute terms, neither topic model induces clusters as accurate in terms of V_1 as a bag-of-words clustering (denoted TF-IDF in Figure 6.12a).

In Figure 6.12b we show the V_1 computed in comparing the topic clusters to a bags-of-words clustering of 25, 50, or 100 clusters. We observe a similar patter in terms of the behavior of V_1 given the algorithm and number of topics as compared to the human labeled classes, which we should at the least expect for the English corpus, given the high V_1 (0.83) for the bags-of-words versus the human labels. In all cases, as with the classification task, the bigram κLDA (κLDA-2) is consistently worse.

We can repeat the comparison between the topic-model induced clusters and a bag-of-words clustering on the low-resource IARPA Babel transcripts and observe similar trends as with the larger Fisher corpora. For the Babel Tagalog, Vietnamese,



(a) Human labeled topic classes.

(b) Bag-of-words clustered labels.

Figure 6.12: Clustering performance with latent topic features.

Zulu, and Tamil training transcripts, we generated reference bag-of-words clusters in the same manner, except for due to the smaller corpus size we looked at cluster sizes of $|C| = \{10, 20, 30, 40\}$. We used the inferred topics from the models trained using $\mathcal{T} = \{50, 100, 150, 200\}$ latent topics to induce clustering based on the most likely latent topic for each document and computed V_1. The clusering evaluation results for each combination are captured in Figure 6.13.

In general the bigram κLDA models give clusterings that are highly dissimilar to the baseline bag-of-words clusters. For the unigram cache-augmented model, the similarity with the bag-of-words relative to standard LDA varies by language. With the exception of Zulu, the clustering performance of the cache-augmented model generally increases with the number of latent topics. The Vietnamese results stand

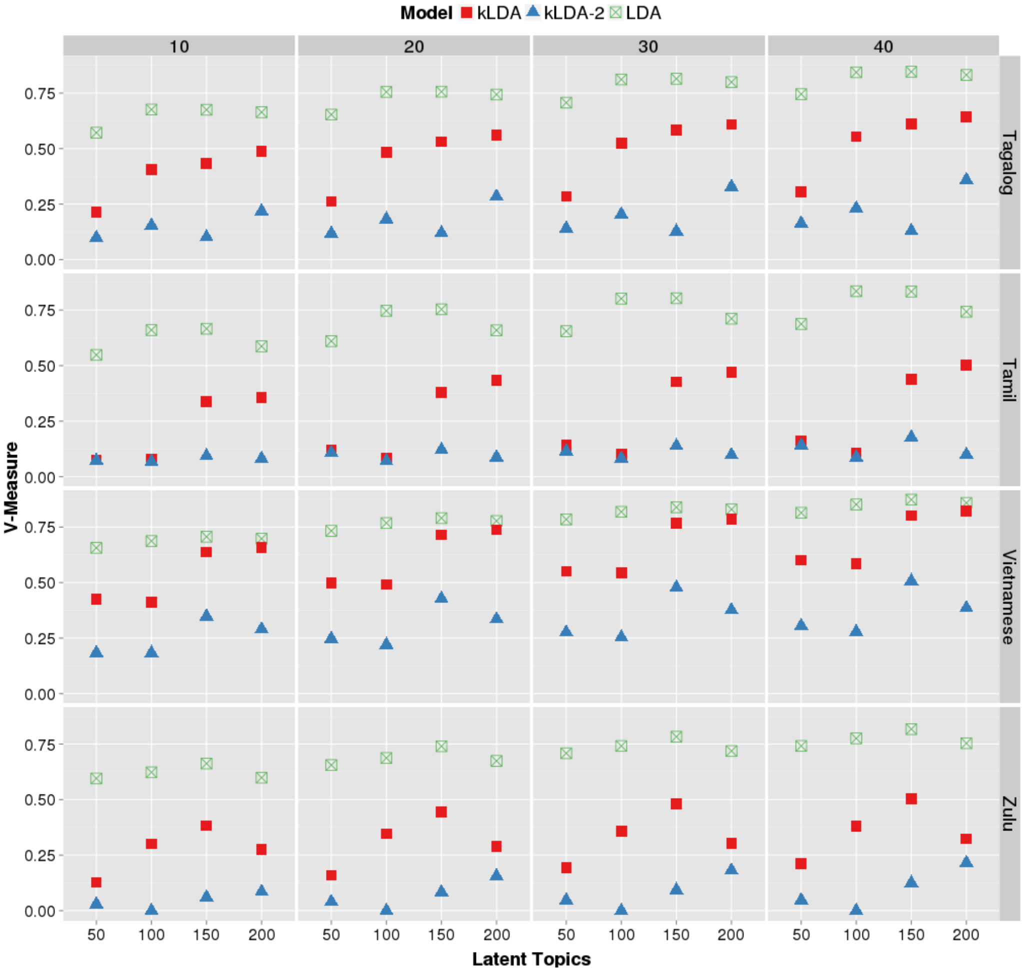


Figure 6.13: Clustering evaluation of babel corpora

out particularly, both in terms of the best performing bigram model and in terms of the unigram κLDA which at 150 and 200 latent topics, appears to generate topic clusters consistent with standard LDA.

In conclusion, we can have some confidence that our proposed topic learns similar topic distributions to standard LDA although they do not prove as effective for classification. The difference in the results between the larger Fisher corpora and the smaller Babel corpora may suggest that the training set size has an effect, but this needs to be separated from language-specific effects.

6.5 Conclusions

In this chapter we analyzed the behavior of a cache-augmented topic model from a variety of perspectives - model likelihood, cache usage and repetition behavior, perplexity, and topic clustering behavior. We considered multiple factors which could affect the various metrics, and different facets of our proposed model responded in different degrees to language properties, training size, model parameters such as the number latent topics, and not surprisingly the intended task for each metric.

In terms of the repetition properties of our proposed models, we observed a number of salient phenomena. We found that the configured number of latent topics impacted the inferred cache usage (as captured by the $\kappa^{(d)}$ estimate) across all languages. We also saw that cache usage aligned with what we might expect given intrinsic morphological and other properties of the particular languages. When we looked at the individual word types that were frequently assigned to the cache state we saw, via English and Spanish examples, that we are not simply replicating word or document frequency properties.

If we focus on the comparison in each case between our proposed model and a standard LDA topic model, we have a mixed set of results in terms of metrics that allow a quantitative comparison such as cluster accuracy, perplexity, or topic classification performance. Perplexity, for example, is lower under our proposed model in two of the 5 low-resource Babel languages, Tagalog and Vietnamese, which by our metrics, also exhibited the most token repetition. Similarly, these two languages exhibited the best performance in terms of clustering accuracy (versus standard LDA) when compared with bag-of-word based clusters.

If we consider how much the task affects interpretation of the model performance, for example, when we compare clustering and classification performance, we want to consider carefully each task and topic model combination. In the next chapter we will do that by looking specifically at external evaluations of our model in the context of speech recognition and retrieval tasks.

Chapter 7

Speech Retrieval

Thus far we have presented a variety of methods to incorporate topic information into the speech retrieval pipeline: topic classification (Chapter 3), repetition-based keyword re-scoring (Chapter4), and an ad-hoc fusion of latent topic and cached N-gram language models (Chapter 4). Building upon the intuitions developed through these experiments, we presented a model in Chapter 5 that formally and distinctly captures both subject matter and repetition aspects of topicality. In this chapter we extrinsically evaluate our proposed model against the spoken keyword retrieval task.

We compare the results from our joint model against the system cascade of re-decoding with topic-only augmented language models followed by re-scoring with a cache-augmented N-gram model. As in Chapter 4 we report our primary results in terms of term-weighted value (TWV) so as to be consistent with published results on the same corpora. Our intent in proposing the model in Chapter 5 was to capture

Language	**Baseline**	**LDA(D)**	**Cache(R)**	**L(D)+C(R)**	**κLDA(D)**
Tagalog	0.244	0.254	0.260	**0.267**	0.266
Vietnamese	0.254	0.269	0.256	**0.271**	**0.271**
Zulu	0.270	0.283	0.276	**0.289**	0.287
Tamil	0.216	0.237	0.229	0.240	**0.241**

Table 7.1: Overall KWS accuracy improvements using joint model (κLDA), compared to LDA and previous cascaded LDA+Cache combination

the same information as in the system combination approach, but for the case where we decode the search corpus with our cache-augmented topic language models, we only perform one additional pass over the data, as opposed to the system cascade which requires two passes. As we summarize in Table 7.1 and subsequently describe in detail, our proposed model performs as well as the system combination approach, but with one less pass over the corpus.

We begin this chapter by reviewing the retrieval task and the corpora involved. Then we elaborate the algorithm by which we incorporate our cache-augmented topic model into the speech recognizer's N-gram language model. In particular we look again at the question of language model interpolation weights. We briefly look at whether sub-document locality, expressed by decaying cache frequencies, is preferable to using the entire document as the local context (for our task it does not). Finally we look at performance on the retrieval task in detail to consider lattice re-scoring versus re-decoding and unigram versus bigram cache models.

7.1 Task and Corpora

The retrieval task is formulated as a term detection or keyword search task, defined by NIST for a 2006 evaluation on English, Mandarin Chinese, and Levantine Arabic broadcast and conversational corpora [7]. The assumption is that document retrieval is dependent on the retrieval of individual keywords. The NIST task focuses on locating a set of key terms (defined as one or more adjacent words) in a corpus of audio. As previously mentioned, the 2006 evaluation also introduced the Term Weighted Value (TWV) metric: given a list of putative term detections, a weighted sum of the false alarm probability and miss probability, averaged over all terms.

We present our empirical retrieval results within the same framework as it is applied to the IARPA Babel retrieval corpora. As in Chapter 4 we focus specifically on the *no target audio reuse* (NTAR) condition for breadth of applicability and to be consistent with other published work on this particular task. This condition states the audio may not be reprocessed after obtaining the search keywords, so it is worth noting that our topic models (or standard LDA) are applied *without any knowledge of the evaluation keyword list.*

As before, we focus on the Limited Language pack (LP) low-resource condition for speech recognition, language, and topic model training. The Limited LP partitions of the Babel corpora contain only 10 hours of transcribed audio and a lexicon restricted to those transcripts. To report recognition (WER) and retrieval (TWV) performance, we decode and search the 10 hour development set, using the released evaluation keywords, again to facilitate comparison with our previous work and other published results. The languages we consider in this chapter include Tagalog, Vietnamese, Zulu and Tamil.[1] These cover the first two years of the Babel program and include the 2013 and 2014 OpenKWS languages [99, 100] (Vietnamese and Tamil).

The ASR acoustic and N-gram language models are the same as those used in Chapter 4 and all experiments carried out within the Kaldi speech recognition toolkit [74]. Kaldi implements language models for ASR as weighted finite state transducers (WFSTs) and relies on the OpenFST [101] package for its language model operations. This has practical implications for implementing custom language models, which we will discuss as we present our full retrieval procedure.

7.2 Procedure

All of the following retrieval experiments follow the basic procedure outline as Algorithm 7.1. In terms of the topic models themselves, we vary the number of latent topics $\mathcal{T}$ and compare the use of a unigram or bigram cache. As with the experiments with standard LDA in Chapter 4 we also compare re-scoring the ASR lattices from the first decode pass to re-decoding the audio with the document-specific, cache-augmented language topic models. We also consider the effect of applying a decay weight to the computation of cache frequencies.

[1] Language collection releases babel106-v0.2g, babel107b-v0.7, babel206b-v0.1e, and babel204b-v1.1b respectively.

Algorithm 7.1 Repetition-based term detection re-scoring

```
Train ASR Acoustic and Language Models
Train Cache-Augmented Topic Language Models
Decode search audio corpus $\mathcal{D}$.
for $d \in \mathcal{D}$ do
  Infer $\theta, \kappa$ from first pass output.
  Compute document-specific unigram model $P_d$ given $\theta^{(d)}$
  for Utterances $u \in d$ do
    Compute cache probabilities from $\hat{u} \neq u$
    Interpolate $P_d$, $P_{cache(u)}$, and $P_{NG}$
    Re-score or Re-decode $u$ to obtain a new lattice.
Perform KWS on new lattices
```

Two primary implementation considerations for this model are: how should the cache probabilities be computed, and how should the topic and cache language models be interpolated with the baseline ASR language model? The cache probabilities need to be computed both during the inference sampling process for $\kappa^{(d)}$ (cf. Equation 5.64) and when augmenting the ASR language model during recognition. As mentioned, we need to select the N-gram **order** of the cache and the **scope** of the cache: how much of the current document ought to influence the cache probabilities.

7.2.1 Cache Frequencies

We pay special attention to the cache scope given the implementation constraints of the WFST framework. The unsmoothed N-gram cache probability computation P_{cache} can be defined according to Equation 7.1, summing over occurrences of the word v and its history H.

$$P_{cache}(w_{d,i} = v | h_i = H) = \frac{\sum_{j=1}^{|d|} \delta(i, j, |d|) \cdot I(w_{d,j} = v \wedge h_j = H)}{\sum_{j=1}^{|d|} \delta(i, j, |d|) \cdot I(h_j = H)} \tag{7.1}$$

We can use this equation to describe a decaying cache by specifying a decay function $\delta(i, j, |d|)$, or a non-decaying cache by letting $\delta(\cdot) = 1$ for all words. For example, a Gaussian decay on the normalized range $[0, 1]$ and parameterized by decay rate λ can be expressed as:

$$\delta(i, j, |d|) = exp\left\{ \frac{-\lambda^2 (i-j)^2}{2|d|^2} \right\} \tag{7.2}$$

Alternatively, the weight function $\delta(\cdot)$ can be used to restrict the cache to a fixed window before and after the current word position:

$$\delta(i, j, |d|) = \begin{cases} 1 & \text{if } |i-j| < 100 \\ 0 & \text{if } |i-j| \geq 100 \end{cases} \tag{7.3}$$

The difficulty applying a cache-based language model within a WFST speech recognition framework is twofold. While a static backoff N-gram language model can be expressed as a WFST, the frequencies of a dynamic cache model change potentially at every word position, particularly if a window or decay function is applied. The cache-based LM cannot be expressed by a single fixed FST.

The second difficulty with applying a dynamic language model is one of efficiency. The WFST decoding system can be expressed as a composition of four WFST components: the language model, G, the lexicon, L, the triphone contexts C, and the

HMM states H. Typically, the system is constructed by composing L with G, then composing with C, and finally with H (cf. [80]).

$$HCLG = H \circ (C \circ (L \circ G)) \tag{7.4}$$

This construction is followed by the Kaldi toolkit and other WFST-based systems. A number of dynamic alternatives have been proposed for re-computing HCLG (cf. [102, 103]), primarily by providing the ability to efficiently compose $(H \circ C \circ L)$ and some G' and obtain the same decoding graph had the original composition order been enforced.

$$HCLG = ((H \circ C \circ L) \circ G') = (H \circ (C \circ (L \circ G')) \tag{7.5}$$

Given these limitations, our approach is to compute a fixed cache on an utterance by utterance basis. In effect, we approximate a fully dynamic cache component. Formally we can define the cache component $P_{cache(u)}$ of Algorithm 7.1 by computing the cache frequencies for Equation 7.1 with either any of the following decay functions: Gaussian (δ_{gauss}), Exponential (δ_{exp}), and windowed (δ_{win}). The function δ_0 is the baseline, non-decaying cache, and $u(i)$ just denotes the utterance containing word w_i.

$$\delta_0(i,j,|d|) = \begin{cases} 1 & \text{if } u(i) \neq u(j) \\ 0 & \text{if } u(i) = u(j) \end{cases} \tag{7.6}$$

$$\delta_{win}(i,j,|d|) = \begin{cases} 1 & \text{if } \frac{|i-j|}{|d|} < \lambda \wedge u(i) \neq u(j) \\ 0 & \text{if } u(i) = u(j) \end{cases} \tag{7.7}$$

$$\delta_{gauss}(i,j,|d|) = \begin{cases} exp\left\{\frac{-\lambda^2(i-j)^2}{2|d|^2}\right\} & \text{if } u(i) \neq u(j) \\ 0 & \text{if } u(i) = u(j) \end{cases} \tag{7.8}$$

$$\delta_{exp}(i,j,|d|) = \begin{cases} exp\left\{-\lambda * |i-j|\right\} & \text{if } u(i) \neq u(j) \\ 0 & \text{if } u(i) = u(j) \end{cases} \tag{7.9}$$

7.2.2 Language Models

Now that we have defined how we will compute the cache probabilities, the second consideration is how to combine the three available language models, the cache, $P_{cache(u)}$, the document-specific topic language model, P_d, and the baseline N-gram model, P_G.

$$P_d(w_i) = \sum_{t=1}^{T} \theta_t^{(d)} \cdot \phi_i^{(t)} \tag{7.10}$$

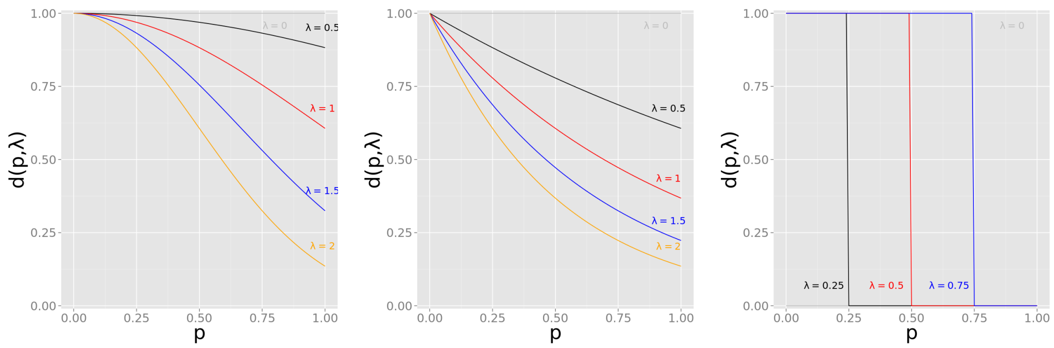
$$P_{dc(u)}(w_i|h_i) = \kappa^{(d)} P_{cache}(w_i|h_i) + (1-\kappa^{(d)}) \cdot P_d(w_i) \tag{7.11}$$

$$P_{Gdc(u)}(w_i) = \lambda P_{dc(u)}(w_i) + (1-\lambda) \cdot P_G(w_i|h_i) \tag{7.12}$$

As we discussed in Chapter 4, the document specific model P_d is computed from each test document given the inferred topic mixture $\theta^{(d)}$ and the topic distributions $\phi^{(t)}$ (cf. Eqn. 7.10). The model we proposed in Chapter 5 models each word as being drawn from either the cache or topic mixture with probability $\kappa^{(d)}$, so we propose the document $\kappa^{(d)}$ as a natural interpolation parameter. Interpreting our model as a unigram language model for a particular utterance u, we obtain $P_{dc(u)}$ according to Equation 7.11.

Lastly, we want to combine the cache-topic mixture with the base N-gram language model P_G. We again use a linear interpolation of probabilities, as with in Chapter 4. Unlike the cache-topic mixture, we have no intuition as to optimal values for the interpolation weight λ, but based on our previous results (cf. [49]) we select the value that minimizes perplexity on the one-best output for that utterance.

We evaluate our approach primarily on keyword retrieval, but we also look at word error rate and lattice recall. As with previous models, re-decoding with the augmented language models consistently improves overall recall of keywords. We can also now show that the cache-augmented topic models when used to re-decode the test corpus, improves retrieval (TWV) and recognition (WER) performance across all languages.



(a) Gaussian (b) Exponential (c) Windowed (δ_{win})

Figure 7.1: Decay function examples

7.3 Results

As previously discussed, we can either use the augmented language models for lattice re-scoring or full re-decoding. We first consider the impact of various decay models as compared to a full document cache (i.e. δ_0) on the re-scoring task. We can show that there is little benefit of applying a decay function to the cache frequencies within each document and for subsequent results we assume no decay in our cache model. We then compile a complete set of results, comparing lattice re-scoring only versus re-decoding with the κLDA augmented language model, and unigram cache versus bigram cache.

7.3.1 Decaying Cache Frequencies

As mentioned, the computation of the cache frequencies can incorporate a decay function $\delta(\cdot)$. In keeping with the notion of locality of repetition, the closer a word

Language ($\mathcal{T}$)	**Baseline**	**Decay**	**TWV**			
	TWV	λ_{win}	1.0[2]	0.75	0.5	0.25
		$\lambda_{gauss,exp}$	0.5	1.0	1.5	2.0
Tagalog (50)	0.244	δ_{win}	**0.257**	0.258	0.258	**0.262**
		δ_{gauss}	0.258	0.258	0.257	0.258
		δ_{exp}	0.257	0.257	0.258	0.258
Vietnamese (200)	0.254	δ_{win}	*0.254*	0.254	0.254	0.255
		δ_{gauss}	0.2540	0.254	0.254	0.254
		δ_{exp}	0.254	0.254	0.253	0.254
Zulu (100)	0.270	δ_{win}	**0.281**	0.281	0.281	0.280
		δ_{gauss}	0.276	0.281	0.281	0.280
		δ_{exp}	0.281	0.281	0.281	0.281
Tamil (100)	0.216	δ_{win}	**0.229**	0.228	0.228	0.226
		δ_{gauss}	0.229	0.228	0.228	0.228
		δ_{exp}	0.228	0.228	0.228	0.228

Table 7.2: TWV effects of applying decaying cache frequencies to lattice re-scoring, compared with the baseline N-gram language model.

occurs to the current utterance, the more it ought to effect the likelihood of the current word. In Chapter 3, application of decay functions to the computation of bags-of-words frequencies had a significant impact on classification tasks. However, when applied to the cache component of the language model for the retrieval or transcription task, we find no significant difference in performance between the decay-weighted cache versus whole-document cache (δ_0).

Table 7.2 shows retrieval performance for lattice re-scoring in terms of the NIST TWV metric for the different decay models over the baseline N-gram model. Table 7.3 shows the transcription performance in terms of WER. We restrict our analysis of the decay-weighted cache models to those models whose number of latent topics

[2] Corresponds to δ_0

Language ($\mathcal{T}$)	**Baseline**	**Decay**	**WER (%)**			
	WER	λ_{win}	1.0	0.75	0.5	0.25
		$\lambda_{gauss,exp}$	0.5	1.0	1.5	2.0
Tagalog (50)	59.8	δ_{win}	59.8	59.8	59.7	59.7
		δ_{gauss}	59.8	59.8	59.7	59.7
		δ_{exp}	59.8	59.7	59.8	59.8
Vietnamese (200)	62.0	δ_{win}	61.9	62.0	62.0	62.0
		δ_{gauss}	62.0	61.9	61.9	62.0
		δ_{exp}	61.9	62.0	62.0	62.0
Zulu (100)	67.6	δ_{win}	67.3	67,2	67.2	67.2
		δ_{gauss}	67.2	67.2	67.2	67.2
		δ_{exp}	67.3	67.2	67.2	67.3
Tamil (100)	75.8	δ_{win}	75.5	75.5	75.5	75.5
		δ_{gauss}	75.5	75.5	75.5	75.5
		δ_{exp}	75.5	75.5	75.5	75.4

Table 7.3: WER effects of applying decaying cache frequencies to lattice re-scoring, compared with the baseline N-gram language model

$\mathcal{T}$ performed best overall in terms of the retrieval task. The λ parameter for the windowed 'decay' function δ_{win} is given in descending order as it is a threshold and not a decay rate, and so moving left to right the cache frequencies are effectively computed from a decreasing fraction of the current document (illustrated graphically in Figure 7.1). Given the limited impact of the various decay functions (in general ¡ 0.1% absolute in each metric), the rest of our experiments are reported with the full non-decaying cache frequencies.

7.3.2 Re-scoring and Re-decoding

When we compare performance between re-scoring and re-decoding using our cache-augmented models, we again observe the same positive effect that we saw in section 4.4.1 on retrieval accuracy (TWV and lattice recall) and for Tamil and Zulu, recognition accuracy (WER). Table 7.5 presents the accuracy results in terms of TWV, Table 7.4 shows results for the same systems in terms of lattice recall (the overall percentage of keyword occurrences captured in the ASR lattices), and Table 7.6 shows the results in terms of WER.

If we look specifically at lattice recall (Table 7.4), we observe 2.5 to 5 % absolute increase in recall in the unigram case (depending on the language), and from 0.7% to 4.6% in the bigram cache case. These results are consistent with the results in Chapter 4 with the topic-only (LDA) augmented models, and supports the premise that by boosting keywords with lower probability under the baseline N-gram model, they survive the decoder pruning low-likelihood paths from the lattice.

We have highlighted the rows for which choice of $\mathcal{T}$ resulted in the highest lattice recall per language (Tagalog:100, Vietnamese:50, Zulu:100, and Tamil:100). However for Tagalog and Vietnamese those choices for $\mathcal{T}$ did not result in the highest overall retrieval accuracy (again, highlighted similarly in Table 7.5).

The bigram cache model (used for re-decoding) consistently underperformed the unigram cache model in terms of lattice recall and in terms of TWV. The difference is not as evident for lattice re-scoring, but it is clear that the bigram cache is not offering any additional benefit.

Retrieval in all languages is improved by decoding with the cache-augmented topic model. With respect to recognition, the results varied widely depending on language.

Language	$\mathcal{T}$	**Baseline**	**Lattice Recall (%)** **1-gram**	 **2-gram**
Tagalog	50	77.8	79.8	79.0
	100		**79.8**	79.3
	150		79.3	79.0
	200		79.3	79.0
Vietnamese	50	55.4	**57.1**	56.6
	100		56.3	56.5
	150		56.5	56.4
	200		56.7	56.1
Zulu	50	71.7	74.2	73.1
	100		**74.3**	73.3
	150		74.1	73.1
	200		74.2	73.1
Tamil	50	57.3	62.3	61.5
	100		**62.3**	61.9
	150		62.6	61.7
	200		62.4	61.8

Table 7.4: Improvements in Lattice Recall when decoding with cache-augmented topic language models.

Language	$\mathcal{T}$	**Re-score**		**Re-decode**	
		1-gram	2-gram	1-gram	2-gram
Tagalog	N-grams: 0.244			Trigger: 0.161	
	50	0.257	0.257	**0.261**	0.258
	100	0.256	0.258	0.258	**0.260**
	150	0.254	0.255	0.257	0.256
	200	0.257	0.253	0.254	0.258
Vietnamese	N-grams: 0.254			Trigger: 0.190	
	50	0.256	*0.253*	0.267	0.260
	100	0.254	0.253	0.265	0.259
	150	0.255	0.256	0.264	0.261
	200	0.254	0.254	**0.269**	0.263
Zulu	N-grams: 0.270			Trigger: 0.192	
	50	0.272	0.272	0.285	0.275
	100	0.281	0.277	**0.287**	0.280
	150	0.277	0.279	0.282	0.278
	200	0.279	0.279	0.284	0.278
Tamil	N-grams: 0.216			Trigger: 0.138	
	50	0.225	0.225	0.240	0.234
	100	0.229	0.226	**0.241**	0.238
	150	0.228	0.223	0.237	0.236
	200	0.228	0.225	0.240	0.237

Table 7.5: Effect on Term Weighted Value (TWV) of applying cache-augmented topic model

Language	$\mathcal{T}$	**Re-score**		**Re-decode**	
		1-gram	2-gram	1-gram	2-gram
Tagalog	N-grams: 60.0		Trigger: 61.7		
	50	59.8	**59.6**	59.8	59.7
	100	60.0	59.8	60.0	59.8
	150	*60.2*	59.9	*60.1*	60.0
	200	*60.3*	60.0	*60.3*	60.1
Vietnamese	N-grams: 62.0		Trigger: 63.7		
	50	61.9	61.8	61.9	61.8
	100	61.9	61.8	61.9	61.9
	150	61.9	61.8	61.9	61.9
	200	61.9	61.8	62.0	61.9
Zulu	N-grams: 67.6		Trigger: 69.2		
	50	67.2	67.2	**67.1**	67.1
	100	67.3	67.2	67.2	67.2
	150	67.5	67.3	67.2	67.2
	200	67.2	67.4	67.2	67.3
Tamil	N-grams: 75.8		Trigger: 76.9		
	50	75.5	75.4	75.5	75.4
	100	75.5	75.5	**75.4**	75.5
	150	75.5	75.4	75.5	75.4
	200	75.6	75.6	75.6	75.6

Table 7.6: Effect on Word Error Rate (%) of applying cache-augmented topic model

The WER of the Tamil and Zulu systems were improved by the ungram models by 0.5% and 0.4% absolute over the baseline, whereas Vietnamese improved by at most 0.2% and Tagalog was actually worsened by up to 0.3% absolute.

Given these full sets of results we conclude by comparing the best results of Table 7.5 with the LDA results from Chapter 4 on a language by language basis (cf. Figure 7.2). We can visualize the impact of adding the local context from the cache probabilities in addition to the latent topic models by comparing the LDA versus κLDA figures. As with our previous work in Chapter 4 we see incremental improvements with the cache information in addition the latent topics in all languages except for Vietnamese. Similarly in Tagalog, we see, as before, the cache information has a proportionally larger effect relative to the latent topics. Nonetheless, we maintain the same conclusion, given the evidence across all four languages, that the topic and cache contexts provide complementary information, effective in boosting keyword retrieval.

7.4 Conclusions

In conclusion, we have demonstrated that our joint cache-augmented topic model captures similar improvements in keyword retrieval to the ad hoc approach described in Chapter 4. By modeling both broad and local contexts in a single model, we arrived at the same result with one fewer passes over the data.

We found no additional benefit from a bigram cache. However, as we have focused

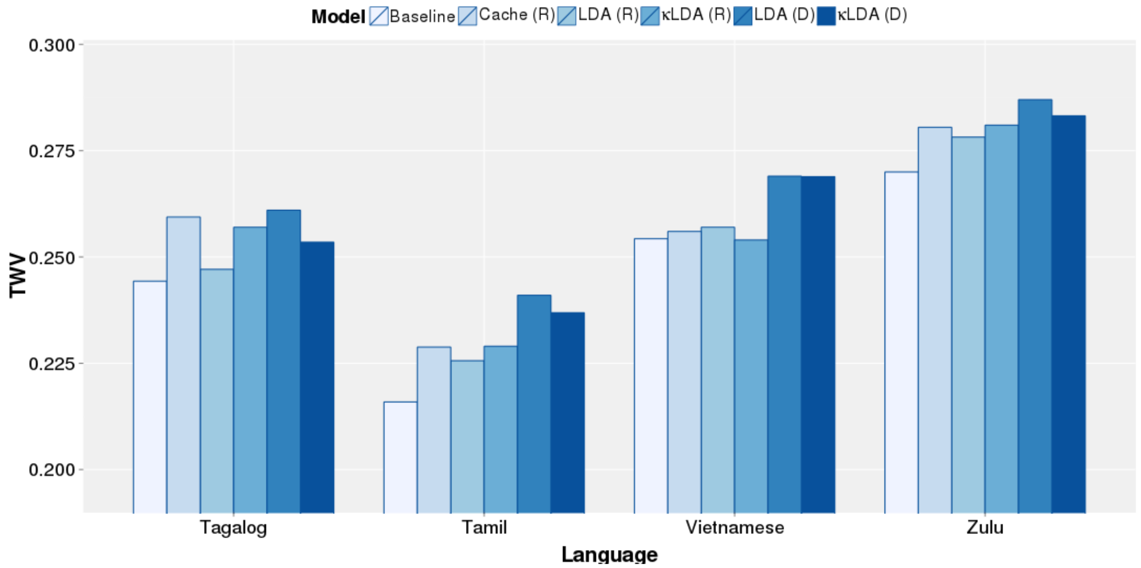


Figure 7.2: Comparison of κLDA retrieval performance with LDA

primarily on the limited resource setting, we intend to extend this work to larger training corpora. As we suggested, bigram cache estimates for more accurate ASR output may in fact be beneficial, however this may be offset by a more accurate N-gram language model overall.

We address our initial constraints of large data volumes and language diversity with a model that improves accuracy in the low resource setting, and we improve computational efficiency without sacrificing retrieval accuracy by moving from a two pass model to a single joint model.

Chapter 8

Summary and Future Work

This thesis has considered the utility of topic information for speech retrieval from a number of different perspectives. In a multinational, interconnected, online, and loquacious world, but with limited and expensive traditional annotated linguistic resources, this thesis demonstrates that anyone in the business of delivering relative spoken content to users benefits by leveraging topic information for both speech recognition and retrieval.

This thesis demonstrates that there is a virtuous cycle between topic information and keyword retrieval. Keyword retrieval drives supervised topic classification of speech, and latent topic information can improve keyword retrieval. This thesis presents a number of novel techniques to exploit these facts to improve speech recognition and retrieval accuracies across a wide range of languages. We conclude this thesis with a summary of the specific contributions described in the preceding chapters, and outline a number of promising future directions for this research.

8.1 Contributions

We can summarize the contributions of this thesis in three main areas. First, we focus on the importance of keywords and locality to topic identification. Second we present novel techniques for exploiting keyword repetition in any language. Third, we develop latent topic and language modelling techniques that jointly leverages broad (subject matter) and local (repetition) topic context to improve both speech recognition and retrieval across a broad range of languages.

8.1.1 Topic Identification

This thesis makes the following contributions in the area of topic identification of spoken documents:

- Quantify the importance of *sufficient* keyword retrieval accuracy over word error rate to predict successful topic identification on ASR output.
- Proposed a new model for discriminative feature selection to add location sensitivity to bag-of-words classification features.

In Chapter 2 we discussed previous work showing the robustness of the topic signal to automatic speech recognition errors (cf. Figure 2.1). In Chapter 3 we further elaborated on the relative insensitivity of the topic signal to recognition errors as expressed in terms of WER (cf. Figure 3.5). We showed that uniformly random word substitutions are significantly more detrimental to topic classification performance than random deletions, but such substitution errors in actual ASR output are not uniformly random. Not only do a small percentage of words from the overall vocabulary contribute to optimal topic classification performance, as has been previously shown, but only a fraction of these keywords need to be recognized correctly to achieve performance similar to what can be achieved with full human transcripts. (cf. Table 3.3).

In addition to the importance of keyword recognition to the classification task, we also demonstrated a strong location-dependent aspect to the topic signal. Following from the concept that topicality is related to local co-occurrence of words, combined with the observation that participants in a conversation tend to drift away from the original (labeled) topic, we incorporate this location sensitivity to bag-of-words feature vectors (cf. Section 3.1). The proposed discriminative, location-sensitive feature vectors out-perform both full document and static topic-drift models.

Given these results, particularly the importance of keyword retrieval to topic classification, the remainder of the thesis focused on applying topic information, and the related phenomena of locality and repetition, to improve keyword retrieval performance.

8.1.2 Repetition in Keyword Retrieval

This thesis makes the following contributions in the modeling of repetition for keyword retrieval:

- Developed a general re-scoring algorithm for applying keyword repetition information to keyword retrieval results from any system in any language.
- In the context of keyword re-scoring, developed a method for computing the score interpolation weight $\widehat{\alpha}$ that generalizes across languages and can be estimated from the *adaptation* statistics of the training data.

Without modifying the underlying speech recognition or retrieval system, we demonstrated that the presence of a high-scoring keyword in a document could be used to boost the scores for subsequent repetitions. In arriving at an effective interpolation formula we also showed how modeling repetition reflects unique language characteristics through the iterpolation weight $\widehat{\alpha}$.

8.1.3 Joint Topic and Repetition Models

This thesis makes the following contributions in topic and language modeling for speech recognition and keyword retrieval:

- Demonstrated the complementary use of *broad* (subject matter) and *local* (repetition) context to improve keyword retrieval in a broad range of languages.

- Presented a new model for jointly modeling both subject-matter and repetition-based latent topics.

- Developed an extensible topic model that captures the related nature of subject-matter topicality and repetition through its latent cache states.

- Showed that latent topics and repetition could be effectively combined in a single joint model that improved speech recognition and keyword retrieval to the same degree as a multi-pass application of individual topic and cache models.

We demonstrated, first in an ad hoc combination, then more formally, how both repetition and subject matter can be expressed in terms of dynamic N-gram language models, and how incorporating those models positively impacts speech recognition and retrieval systems. In isolation we showed that either *broad topic context* (subject matter) or *local context* (as captured by within-document N-gram repetition) could be added to N-gram languages models to improve repetition. The magnitude of the effect of either type of topic information depends on language specific characteristics. However we have shown that together, the two types of topic information are *complementary* in terms of improving speech retrieval in all languages considered.

Based on this result we showed that we can jointly capture the two phenomena with a single model, with positive results both in terms of *intrinsic* analysis of spoken corpora and in terms of *extrinsic*, task-based, retrieval results. Our model captures properties of word repetition for each corpus under consideration different from traditional frequency-based metrics and demonstrates language-specific behavior consistent with known properties of the languages considered. When incorporated into speech recognition systems, we can demonstrate on a spectrum of spoken language corpora that our proposed model improves both overall speech recognition accuracy as well as keyword retrieval accuracy.

Finally, when we contrast our ad hoc pipeline from Chapter 4 with our formal model from Chatper 5, we show that we can achieve equivalent performance improvements, by incorporating both repetition and subject matter, but with one rather than two additional passes over the audio.

8.2 Future Work

We would suggest that the line of work described here in this thesis can be extended in multiple directions: first in terms of generalization and further consideration of the models presented in this work, and second in terms of further development of the models from the perspective of computational efficiency. We feel that broad applicability, in terms of languages to which they are successfully applied, of the concepts discussed in this thesis warrants further exploration of the means by which they might become viable in commercial production systems.

In terms of our proposed cache-augmented topic language models, there are many probabilistic topic model frameworks to which we could consider the addition of explicit cache or repetition behavior. As an example, it would be reasonable to consider our model under other, more general Dirichlet process or hierarchical topic model frameworks. We would ask whether the repetition behavior we modeled explicitly might or might not arise naturally and separate from subject-matter topics under other models. Additionally, we would like to consider frameworks which could efficiently represent N-gram topic mixtures in addition to an N-gram cache.

In terms of efficiency, we highlight two complementary directions for future work. First, and most straightforward, we would incorporate known Finite State Transducer (FST) composition algorithms designed specifically for using dynamic language models with a WFST-based ASR system such as Kaldi. For experimental purposes we re-constructed the ASR decoding graph for each segment's topic-cache-augmented model P_{Ldc}. In a production setting, on-the-fly graph construction techniques such as proposed by Allauzen et al. [103], suggest our dynamic language model approach could be efficiently applied.

Secondly, we would examine techniques to speed up estimation of our model parameters for topic and cache usage at the point where new audio is to be decoded. Indeed with the recent expansion of neural-net (NN) based language models, a natural extension of this work would be to ask what other methods could be used to approximate the topic and cache estimates, θ and κ, necessary for generating document-specific cache language models as described in Chapter 5-7. In particular we would envision comparing our approach with techniques such as Recurrent Neural Network

Language models (RNNLMs) or Long-Short-Term Memory language models (LSTMs) which also aim to capture context beyond simple N-grams.

Additionally, in neural net acoustic modeling, there is some evidence that output layers representing context-dependent triphone likelihoods (referred to as *senones*) capture lexical as well as purely acoustic content (cf. [104]). These likelihoods are produced without a full decoding pass from the ASR system and could be used in approximating topic information before lattice generation, resulting in a single pass system.

There are many opportunities for efficiently exploiting topic and repetition for speech recognition and retrieval which we have not listed here. It is our belief that this will continue to be a rich and widely applicable source of gains for speech processing systems in any language.

> What we call the beginning is often the end
> And to make an end is to make a beginning.
> The end is where we start from. And every phrase
> And sentence that is right (where every word is at home,
> Taking its place to support the others,
> The word neither diffident nor ostentatious,
> An easy commerce of the old and the new,
> The common word exact without vulgarity,
> The formal word precise but not pedantic,
> The complete consort dancing together)
> Every phrase and every sentence is an end and a beginning,
> Every poem an epitaph.
>
> T.S. Eliot, Little Gidding

Appendix A

Topic Drift Gradient Descent Derivation

We present a method for estimating $\widehat{\lambda}_w$ for each word w in the classification training corpus using a minimum classification error (MCE) training [105], following a derivation for the gradient-descent update given by [19]. The major difference between derivations is that the parameter λ_w we wish to optimize occurs inside the decay function, so we are obliged to take the partial derivative of the decay, $d(p, \lambda_w)$ as well as of the Naive Bayes scoring function.

The MCE method attempts to minimize a loss function $l(D)$ over the training corpus, where l is defined for each document D. The loss function is defined in terms of a misclassification measure $M(D)$, the same measure as in [19], and loss function $l(D)$, which maps $M(D)$ to a $[0, 1]$ range. Here t_C is the correct topic label for D and

and t_I is the incorrect topic with the highest score. For notational simplicity, we also define $M(w)$ as the word specific component of $M(D)$.

$$M(D) = S(t_I|D) - S(t_C|D) \tag{A.1}$$

$$l(D) = \frac{1}{1 + e^{-\beta M(D)}} \tag{A.2}$$

$$M(w) = log\left(\frac{P(w|t_I)}{P(w|\overline{t_I})}\right) - log\left(\frac{P(w|t_C)}{P(w|\overline{t_C})}\right) \tag{A.3}$$

The scoring function is defined by combining a log-likelihood ratio form of Naive Bayes (Equation A.4) with the decay-weighted counts (Equation A.5. We obtain an $S(t|D)$ where the per-word contribution is a weighted sum over each position, rather than a single term (Equation A.6).

$$S(t|D) = \sum_{w \in D} c_w \cdot log\left(\frac{P(w|t)}{P(w|\overline{t})}\right) \tag{A.4}$$

$$c_w = \sum_{i=1}^{|D|} d(\frac{i}{|D|}, \widehat{\lambda}_w) \cdot I_w(w_i) \tag{A.5}$$

$$S(t|D) = \sum_{i}^{|D|} d(\frac{i}{|D|}, \lambda_w) \cdot log\left(\frac{P(w_i|t)}{P(w_i|\overline{t})}\right) \tag{A.6}$$

We now compute the partial derivative and update equations for gradient-descent optimization of $M(D)$, which contain our decay function $d(p, \lambda_w)$.

$$\frac{\partial l(D)}{\partial \lambda_w} = \beta \cdot l(D)(1 - l(D)) \cdot M(w) \left[\sum_{i}^{|D|} \frac{\partial d(\frac{i}{|D|}, \lambda_w)}{\partial \lambda_w} \right] \tag{A.7}$$

$$\lambda_w{}' = \lambda_w - \epsilon \frac{1}{N} \sum_{j=1}^{N} \frac{\partial l(D_j)}{\partial \lambda_w} \tag{A.8}$$

Given the computational cost of performing the gradient descent, we evaluate the MCE training using only the exponential and Gaussian decay functions, which performed better in our static tests. The partial derivatives for $d(p, \lambda_w)$ are given as follows:

$$\frac{\partial d_{exp}}{\partial \lambda_w} = -p \cdot exp\,(-\lambda_w \cdot p) \tag{A.9}$$

$$\frac{\partial d_{gauss}}{\partial \lambda_w} = -p^2 \cdot \lambda_w{}^2 \cdot exp\left(-\frac{\lambda_w{}^2 \cdot p^2}{2}\right) \tag{A.10}$$

In our experiments we use 5-fold cross-validation to compute the training loss. We found empirically that for the English data $\epsilon = 10$ and $\beta = 0.01$ achieved the best results, whereas for Spanish $\epsilon = 100$ and $\beta = 0.1$ were best.

Appendix B

Gibbs Sampler Pseudocode

Here we present pseudocode for the implementation of the Gibbs sampler derived in Section 5.2. For practical reasons, we maintain the cache on an utterance by utterance basis. Cache probabilities are conditioned on counts from all utterances in the document except the one whose states are currently being sampled.

Algorithm B.1 Sampler initialization

```
Initialize $K$ and $Z$ states randomly
for all $t \in \mathcal{T}$ do
    for all $v \in V$ do
        $C_v^t = \sum_{i,d} I(w_{d,i} = v \cap z_{d,i} = t)$
    $F_t = \sum_v C_v^t$
```

The following code snippets are repeated n times for each document d, where n is the overall number of sampling iterations.

Algorithm B.2 Per-document initialization - each iteration over d

```
for all $u_j \in d$ do
    Add n-grams in $u$ to cache
$L_{d_0} = \sum_i I(k_{d,i} = 1)$
$T_{d_0} = \sum_i I(k_{d,i} = 0)$
for all $t \in \mathcal{T}$ do
    $N_t = \sum_i I(z_{d,i} = t)$
for all $t \in \mathcal{T}$ do
    for all $v \in V$ do
        $C_v^t = \sum_i I(w_{d,i} = v \cap z_{d,i} = t)$
    $F_t = \sum_v C_v^t$
```

We then sample states K and Z one utterance (or sentence) at a time. We first sample all K_u states, then Z_u states for the current utterance u.

Algorithm B.3 Single iteration - K_u

```
Remove n-grams in $u$ from cache
for all $w_{d,i} \in u$ do
    $k_{old} = k_{d,i}$
    if $k_{old} = 1$ then
        $L_{d_0} = L_{d_0} - 1$                                    # Decrement cache state count
        $s_0 = (\nu_1 + T_{d_0})$                              # Sampler proportional mass for $k = 0$
    else
        $T_{d_0} = T_{d_0} - 1$                                    # Decrement topic state count
        $z_{old} = z_{d,i}$                                   # Decrement topic count variables
        $C_{w_{d,i}}^{z_{old}} = C_{w_{d,i}}^{z_old} - 1$
        $F_{z_{old}} = F_{z_old} - 1$, $N_{z_old} = N_{z_old} - 1$
        $s_0 = (\nu_1 + T_{d_0}) \cdot (\beta + C_{w_{d,i}}^{(z_{d,i})}) / (\beta \cdot |V| + F_t)$
    $s_1 = P_{cache}(w_{d,i}) \cdot (\nu_0 + L_{d_0})$
    Draw  $s \sim Uniform(0, s_0 + s + 1)$
    if $s > s_0$ then
        $k_{d,i} = 1$, $L_{d_0} = L_{d_0} + 1$
    else
        $k_{d,i} = 0$, $T_{d_0} = T_{d_0} + 1$
```

Algorithm B.4 Single iteration - Z_u

```
for all $w_{d,i} \in u$ where $k_{d,i} = 0$ do
    for all $t \in \mathcal{T}$ do
        $s_t = (\alpha_t + N_t + 1) \cdot (\beta + C^t_{w_{d,i}} + 1)/(\beta \cdot |V| + F_t + 1)$
    Draw  $s \sim Uniform(0, \sum_t s_t)$
    $s_0 = 0$
    for all $t \in \mathcal{T}$ do
        $s_0 = s_0 + s_t$
        if $s < s_0$ then
            $z_{d,i} = t$                               # New sampled topic is now $t$
            $C^t_{w_{d,i}} = C^t_{w_{d,i}} + 1$         # Increment topic count variables
            $F_t = F_t + 1,\ N_t = N_t + 1$
Add n-grams in $u$ back to cache
```

Appendix C

Model Convergence

Here we include log-likelihood convergence figures for all of the languages considered in Chapter 6, followed by convergence figures for κ estimates in all languages.

Corpus	**Topics**	**LDA**	**κLDA-1**	**κLDA-2**
Turkish	50	-7.497 (0.02)	-9.239 (0.03)	-8.351 (0.03)
	100	-7.554 (0.02)	-9.173 (0.04)	-8.535 (0.02)
	150	-7.554 (0.03)	-9.022 (0.04)	-8.620 (0.03)
	200	-7.543 (0.04)	-8.865 (0.04)	-8.681 (0.03)
Tagalog	50	-6.551 (0.02)	-8.288 (0.03)	-7.761 (0.02)
	100	-6.523 (0.02)	-8.210 (0.03)	-7.910 (0.04)
	150	-6.525 (0.01)	-8.081 (0.04)	-7.952 (0.04)
	200	-6.508 (0.02)	-7.898 (0.04)	-7.919 (0.03)
Vietnamese	50	-6.564 (0.03)	-8.352 (0.03)	-8.067 (0.02)
	100	-6.498 (0.01)	-8.245 (0.03)	-8.044 (0.03)
	150	-6.483 (0.01)	-8.017 (0.03)	-8.019 (0.02)
	200	-6.471 (0.01)	-7.788 (0.03)	-7.942 (0.04)
Zulu	50	-7.864 (0.04)	-9.758 (0.04)	-8.418 (0.03)
	100	-7.887 (0.02)	-9.912 (0.03)	-8.594 (0.03)
	150	-7.881 (0.02)	-9.855 (0.03)	-8.715 (0.03)
	200	-7.910 (0.02)	-9.787 (0.05)	-8.844 (0.03)
Tamil	50	-8.044 (0.03)	-9.919 (0.04)	-8.629 (0.03)
	100	-7.887 (0.02)	-9.993 (0.04)	-8.853 (0.03)
	150	-8.048 (0.01)	-9.869 (0.04)	-8.991 (0.04)
	200	-7.910 (0.02)	-9.761 (0.04)	-9.063 (0.03)
Spanish (CallHome)	50	-6.981 (0.03)	-8.439 (0.03)	-7.933 (0.04)
	100	-7.034 (0.02)	-8.341 (0.04)	-8.164 (0.04)
	150	-6.994 (0.03)	-8.228 (0.05)	-8.240 (0.06)
	200	-6.971 (0.03)	-8.074 (0.03)	-8.227 (0.04)
Spanish (Fisher)	50	-7.431 (0.02)	-8.384 (0.23)	-8.193 (0.27)
	100	-7.505 (0.01)	-8.381 (0.01)	-8.270 (0.03)
	150	-7.553 (0.01)	-8.341 (0.02)	-8.329 (0.03)
	200	-7.544 (0.01)	-8.292 (0.02)	-8.436 (0.03)

Table C.1: Overall Log-likelihood and sample standard deviation per word

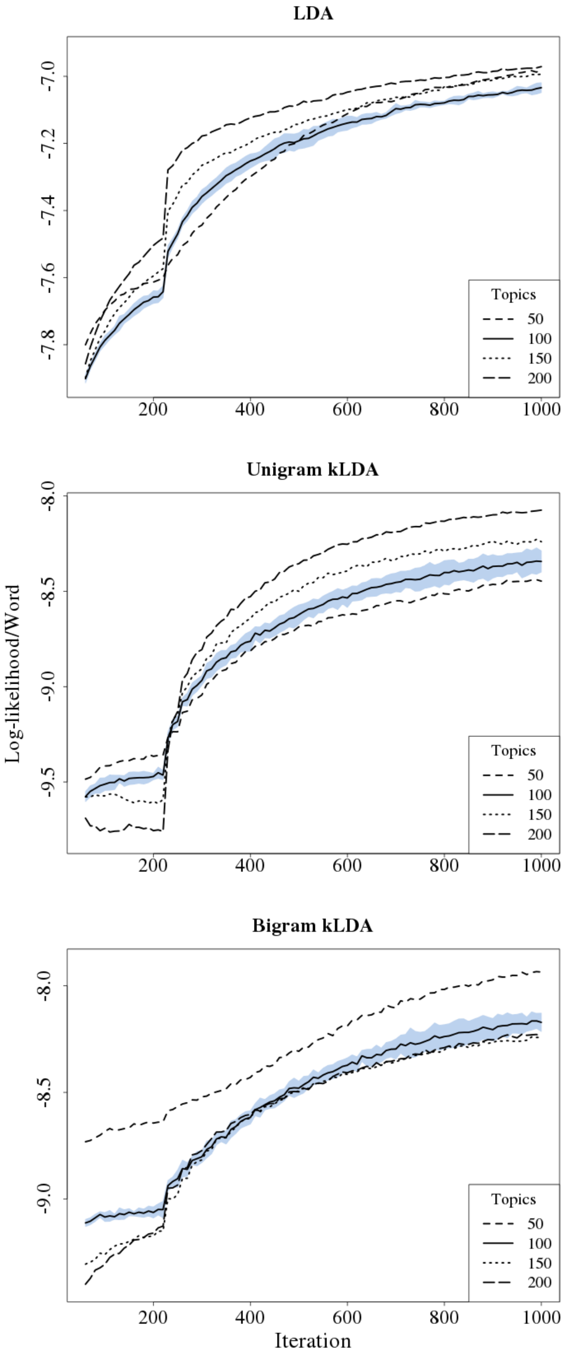


Figure C.1: Model log-likelihood convergence over sampling process, CallHome Spanish

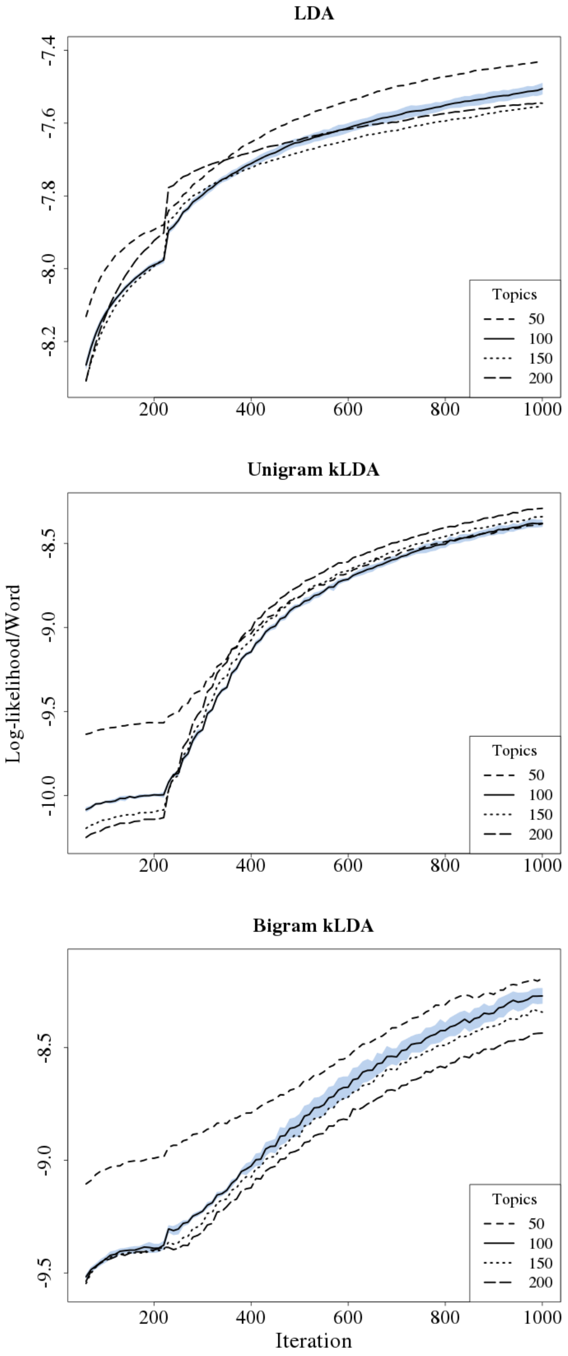


Figure C.2: Model log-likelihood convergence over sampling process, Fisher Spanish

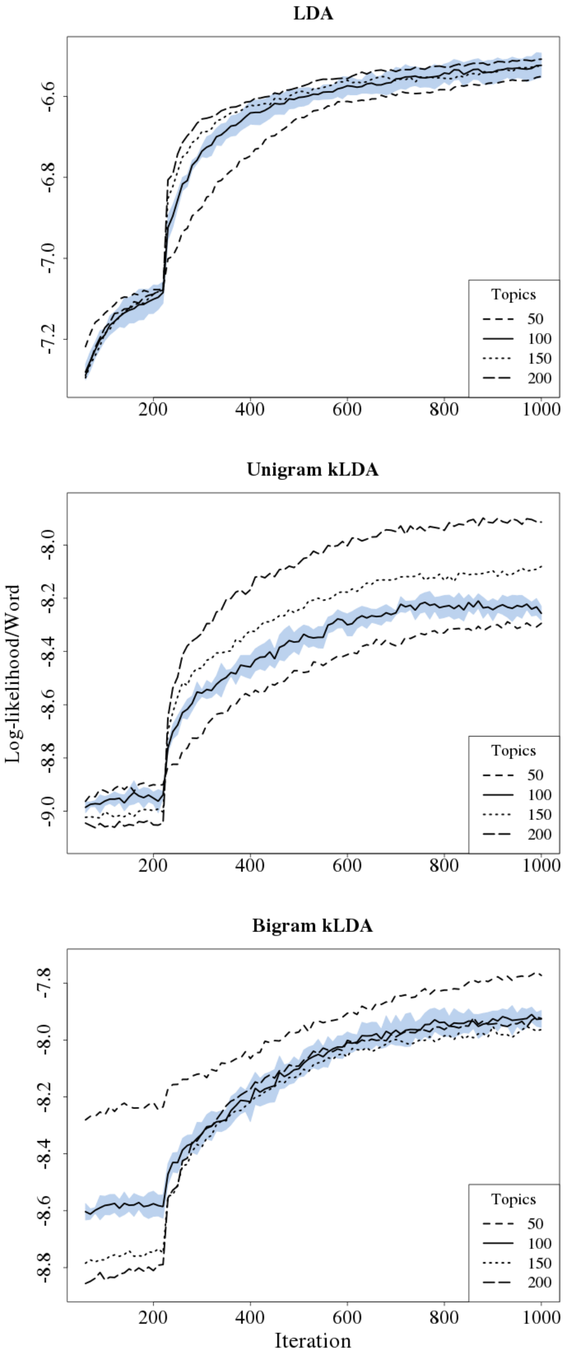


Figure C.3: Model log-likelihood convergence over sampling process, Tagalog

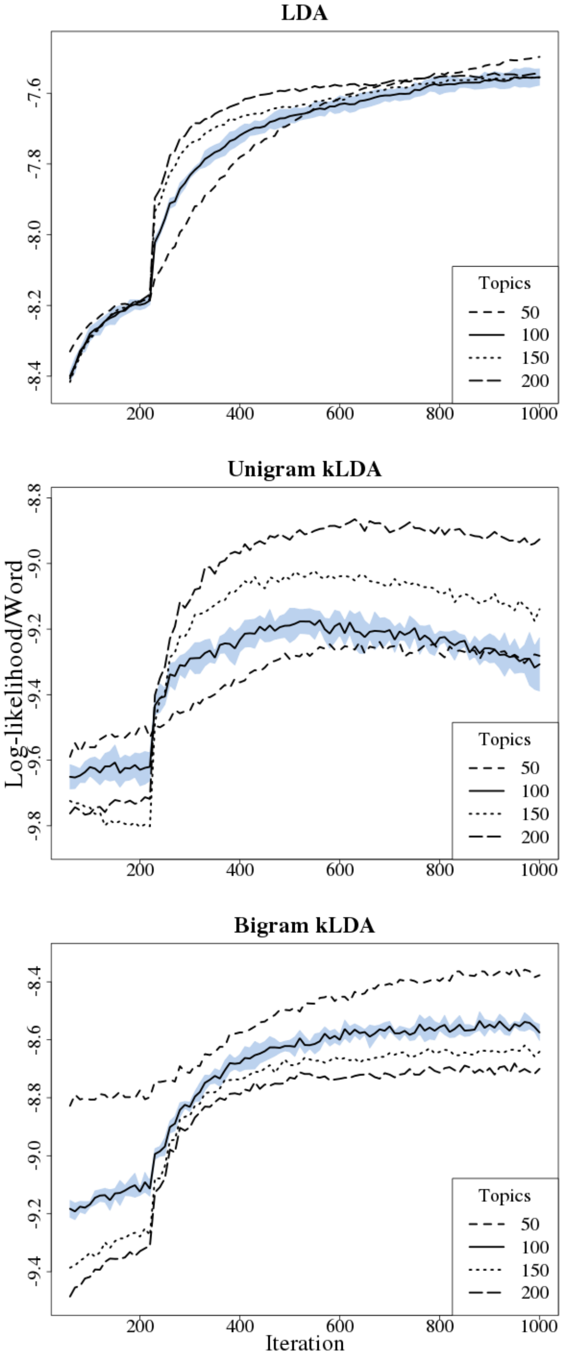


Figure C.4: Model log-likelihood convergence over sampling process, Turkish

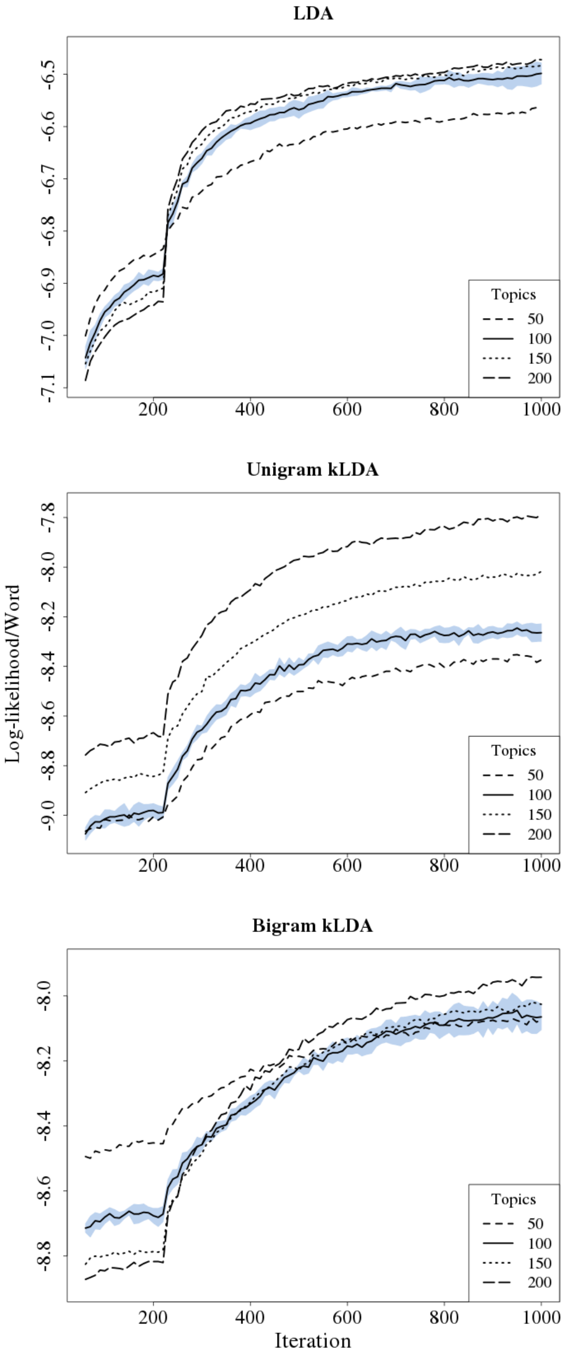


Figure C.5: Model log-likelihood convergence over sampling process, Vietnamese

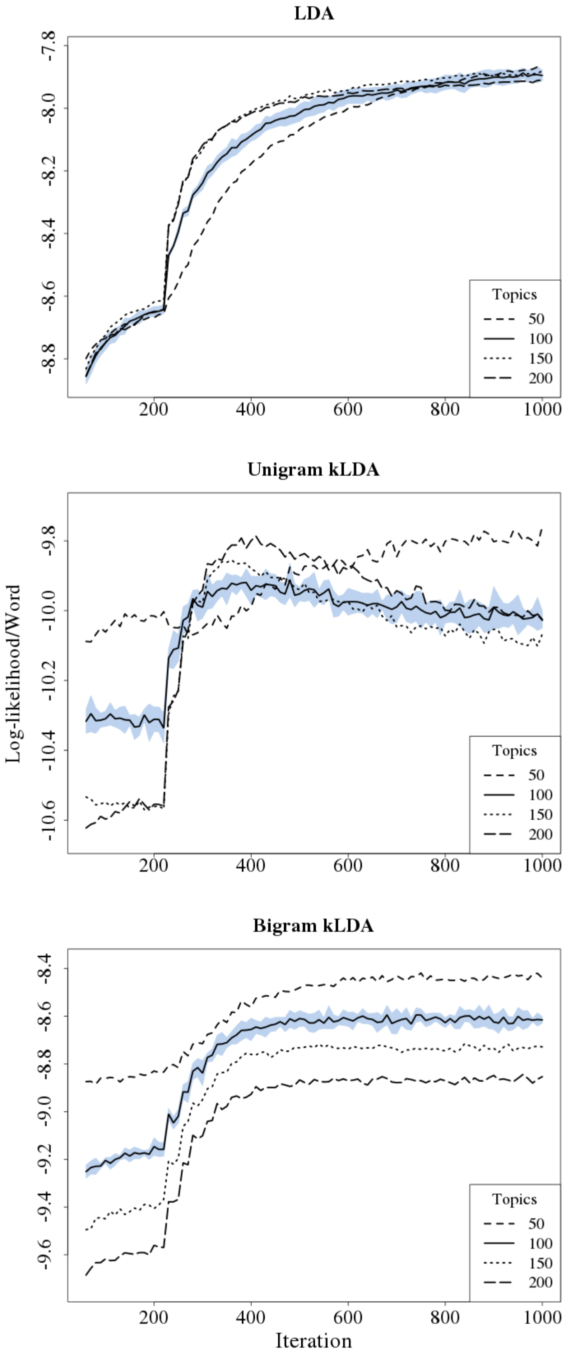


Figure C.6: Model log-likelihood convergence over sampling process, Zulu

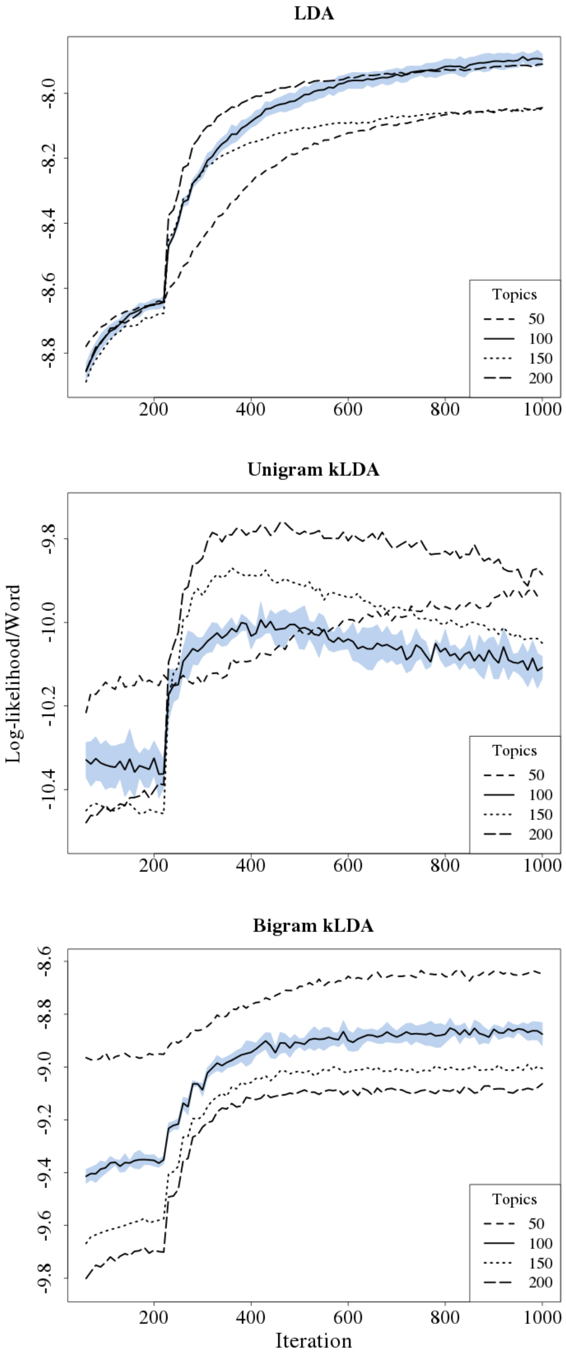
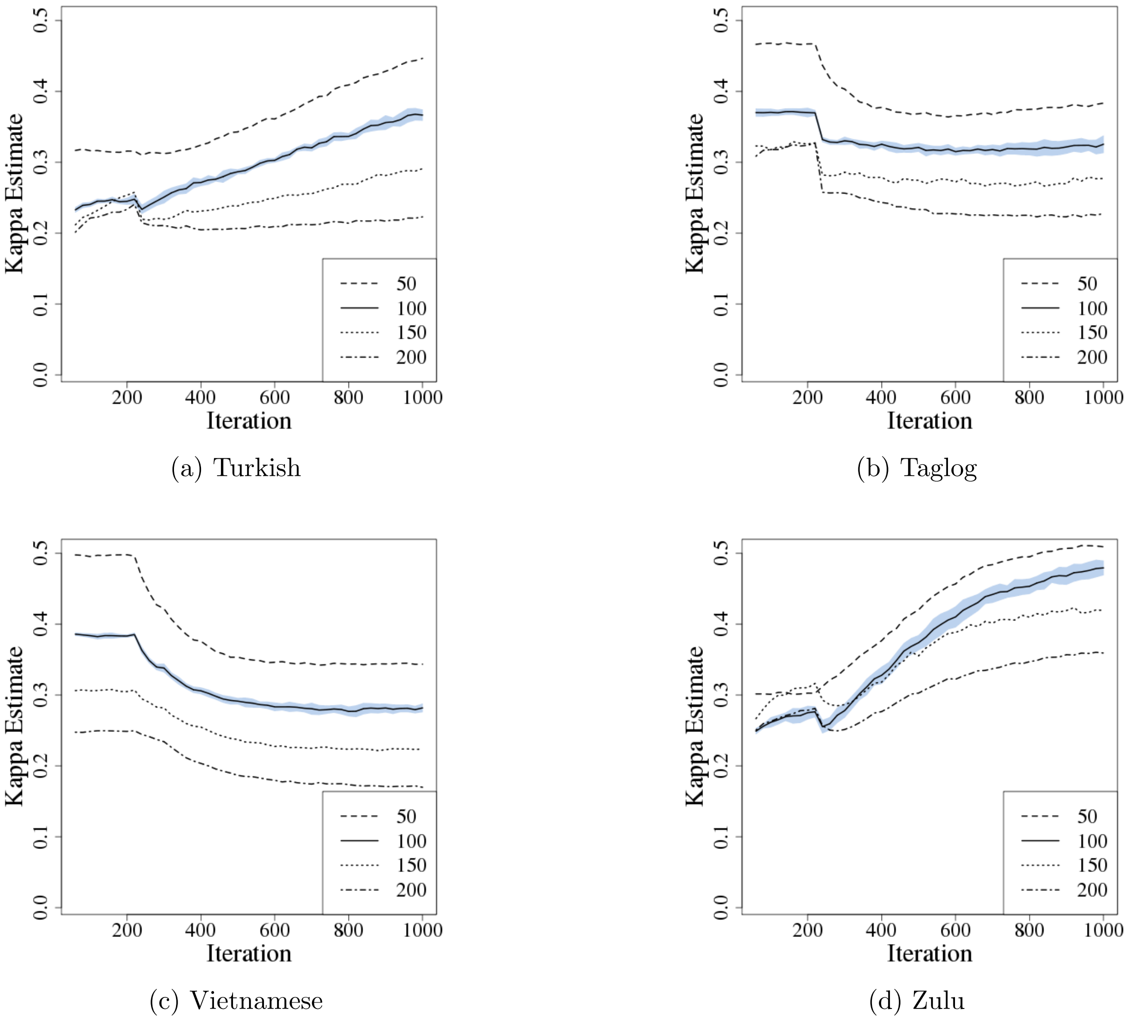


Figure C.7: Model log-likelihood convergence over sampling process, Tamil



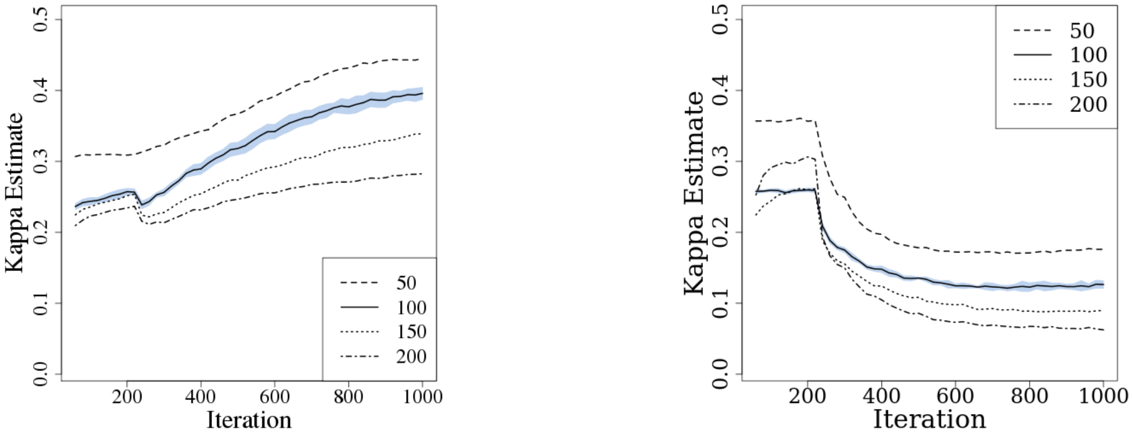
(a) Turkish

(b) Taglog

(c) Vietnamese

(d) Zulu

Figure C.8: κ convergence in κLDA sampling process.



(a) Tamil

(b) CallHome Spanish

Figure C.9: κ convergence in κLDA sampling process.

Bibliography

[1] YouTube, "Statistics - YouTube," http://www.youtube.com/yt/press/statistics.html, [Online; accessed Sep-2015].

[2] Google, "Google Mission Statement," http://www.google.com/about/company/, [Online; accessed Sep-2015].

[3] J. Wintrode, "Limited Resource Term Detection For Effective Topic Identification of Speech," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[4] P. N. Howard, "The Arab Spring's Cascading Effects," http://www.psmag.com/navigation/politics-and-law/the-cascading-effects-of-the-arab-spring-28575/, Feb 2011, [Online; accessed Sep-2015].

[5] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval.* ACM Press, 1999.

[6] J. S. Garofolo, C. G. Auzanne, and E. M. Voorhees, "The TREC Spoken Document Retrieval Track: A success story," *NIST Special Publication SP*, no. 246, pp. 107–130, 2000.

[7] NIST, "The Spoken Term Detection (STD) 2006 Evaluation Plan," `http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf`, 2006, [Online; accessed Sep-2015].

[8] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 Spoken Term Detection Evaluation," in *Proc. SIGIR*, 2007, pp. 51–57.

[9] M. Harper, "IARPA Solicitation IARPA-BAA-11-02," `http://www.iarpa.gov/solicitations_babel.html`, 2011, [Online; accessed Sep-2015].

[10] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, Mar. 2002.

[11] T. J. Hazen, "Topic Identification," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, pp. 319–356, 2011.

[12] A. McCallum, K. Nigam *et al.*, "A Comparison of Event Models for Naive Bayes Text Classification," in *AAAI-98 Workshop on Learning for Text Categorization*, 1998, pp. 41–48.

[13] J. D. Rennie, L. Shih, J. Teevan, D. R. Karger *et al.*, "Tackling the Poor

Assumptions of Naive Bayes Text Classifiers," in *Proc. of the International Conference on Machine Learning (ICML)*, 2003.

[14] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech Corpus for Research and Development," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 1992, pp. 517–520.

[15] B. Peskin *et al.*, "Improvements in Switchboard Recognition and Topic Identification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 1996, pp. 303–306.

[16] T. J. Hazen, F. Richardson, and A. Margolis, "Topic Identification from Audio Recordings Using Word and Phone Recognition Lattices," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU).* IEEE, 2007, pp. 659–664.

[17] J. Wintrode and S. Kulp, "Techniques for Rapid and Robust Topic Identification of Conversational Telephone Speech," in *Proc. of Interspeech*, 2009.

[18] G. Chen, S. Khudanpur, D. Povey, J. Trmal, D. Yarowsky, and O. Yilmaz, "Quantifying the Value of Pronunciation Lexicons for Keyword Search in Low Resource Languages," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[19] T. J. Hazen and A. Margolis, “Discriminative Feature Weighting using MCE Training for Topic Identification of Spoken Audio Recordings,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 4965–4968.

[20] M. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, “Unsupervised Training of an HMM-based Self-Organizing Unit Recognizer with Applications to Topic Classification and Keyword Discovery,” *Computer Speech & Language*, 2013.

[21] M. Dredze, A. Jansen, G. Coppersmith, and K. Church, “NLP on Spoken Documents Without ASR,” in *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, 2010, pp. 460–470.

[22] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, “Rapid Evaluation of Speech Representations for Spoken Term Discovery,” in *Proc. of Interspeech*, 2011, pp. 821–824.

[23] M. A. Hearst and C. Plaunt, “Subtopic Structuring for Full-length Document Access,” in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 1993, pp. 59–68.

[24] M. A. Hearst, “TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages,” *Computational linguistics*, vol. 23, no. 1, pp. 33–64, 1997.

[25] J. C. Reynar, “Statistical Models for Topic Segmentation,” in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics.* Association for Computational Linguistics, 1999, pp. 357–364.

[26] F. Y. Choi, “Advances in Domain Independent Linear Text Segmentation,” in *North American chapter of the Association for Computational Linguistics conference.* Association for Computational Linguistics, 2000, pp. 26–33.

[27] H. M. Wallach, “Topic Modeling: Beyond Bag-of-words,” in *Proceedings of the 23rd international conference on Machine learning.* ACM, 2006, pp. 977–984.

[28] L. Du, W. Buntine, and M. Johnson, “Topic Segmentation with a Structured Topic Model,” in *Proceedings of NAACL-HLT*, 2013, pp. 190–200.

[29] V.-A. Nguyen, J. Boyd-Graber, and P. Resnik, “SITS: A Hierarchical Non-parametric Model using Speaker Identity for Topic Segmentation in Multiparty Conversations,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1.* Association for Computational Linguistics, 2012, pp. 78–87.

[30] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, “The ICSI Meeting Corpus,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1. IEEE, 2003, pp. I–364.

[31] Ries, Levin, Valle, Lavie, and Waibel, "Shallow Discourse Genre Annotation in CallHome Spanish," *Proceedings of the Second International Conference On Language Resources And Evaluation*, 2000.

[32] J. Wintrode, "Leveraging Locality for Topic Identification of Conversational Speech," in *Proc. of Interspeech*, 2013.

[33] M. Morchid, G. Linares, M. El-Beze, and R. De Mori, "Theme Identification in Telephone Service Conversations using Quaternions of Speech Features," in *Proc. of Interspeech*, 2013.

[34] F. Jelinek, B. Merialdo, S. Roukos, and M. Strauss, "A Dynamic Language Model for Speech Recognition," in *HLT*, vol. 91, 1991, pp. 293–295.

[35] R. Kuhn and R. De Mori, "A Cache-based Natural Language Model for Speech Recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 6, pp. 570–583, 1990.

[36] R. Rosenfeld, "Adaptive Statistical Language Modeling: A Maximum Entropy Approach," Ph.D. dissertation, CMU, 1994.

[37] N. Singh-Miller and C. M. Collins, "Trigger-based language modeling using a loss-sensitive perceptron algorithm," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2007.

[38] S. F. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for

Language Modeling," in *Proceedings of the 34th annual meeting on Association for Computational Linguistics.* Association for Computational Linguistics, 1996, pp. 310–318.

[39] D. Beeferman, A. Berger, and J. Lafferty, "A Model of Lexical Attraction and Repulsion," in *Proc. of EACL.* Association for Computational Linguistics, 1997, pp. 373–380.

[40] ——, "Statistical Models for Text Segmentation," *Machine Learning*, vol. 34, no. 1-3, pp. 177–210, 1999.

[41] R. Florian and D. Yarowsky, "Dynamic Nonlocal Language Modeling via Hierarchical Topic-based Adaptation," in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics.* Association for Computational Linguistics, 1999, pp. 167–174.

[42] S. Khudanpur and J. Wu, "A Maximum Entropy Language Model Integrating N-grams and Topic Dependencies for Conversational Speech Recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1. IEEE, 1999, pp. 553–556.

[43] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *the Journal of Machine Learning Research (JMLR)*, vol. 3, pp. 993–1022, 2003.

[44] T. Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine Learning*, vol. 42, no. 1, pp. 177–196, 2001.

[45] A. Heidel, H.-a. Chang, and L.-s. Lee, "Language Model Adaptation using Latent Dirichlet Allocation and an Efficient Topic Inference Algorithm," in *Proc. of Interspeech*, 2007.

[46] B.-J. P. Hsu and J. Glass, "Style & Topic Language Model Adaptation using HMM-LDA," in *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP).* Association for Computational Linguistics, 2006, pp. 373–381.

[47] Y. Liu and F. Liu, "Unsupervised Language Model Adaptation via Topic Modeling Based on Named Entity Hypotheses," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2008, pp. 4921–4924.

[48] S. Huang and S. Renals, "Unsupervised Language Model Adaptation Based on Topic and Role Information in Multiparty Meetings," in *Proc. of Interspeech*, 2008.

[49] J. Wintrode and K. Sanjeev, "Combining Local and Broad Topic Context to Improve Term Detection," in *Proc. of IEEE Spoken Language Technology Workshop*, 2014.

[50] J. M. Ponte and W. B. Croft, "A Language Modeling Approach to Information Retrieval," in *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval.* ACM, 1998, pp. 275–281.

[51] 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 and development in information retrieval.* ACM, 2006, pp. 178–185.

[52] B. Chen, "Latent Topic Modelling of Word Co-occurence Information for Spoken Document Retrieval," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2009, pp. 3961–3964.

[53] X. Liu and W. B. Croft, "Cluster-based Retrieval using Language Models," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2004, pp. 186–193.

[54] K. Church and W. Gale, "Inverse Document Frequency (IDF): A Measure of Deviations from Poisson," in *Natural language processing using very large corpora.* Springer, 1999, pp. 283–295.

[55] K. W. Church and W. A. Gale, "Poisson Mixtures," *Natural Language Engineering*, vol. 1, no. 2, pp. 163–190, 1995.

[56] K. W. Church, "Empirical Estimates of Adaptation: the Chance of Two Noriegas is Closer to $p/2$ than p^2," in *Proceedings of the 18th conference on Compu-*

tational Linguistics, vol. 1. Association for Computational Linguistics, 2000, pp. 180–186.

[57] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," in *Proc. of the International Conference on Machine Learning (ICML)*, 1997, pp. 412–420.

[58] D. Blackwell and J. B. MacQueen, "Ferguson distributions via polya urn schemes," *Annals of Statistics*, vol. 1, no. 2, pp. 353–355, 03 1973.

[59] T. Minka, "Estimating a Dirichlet Distribution," 2000.

[60] R. E. Madsen, D. Kauchak, and C. Elkan, "Modeling Word Burstiness using the Dirichlet Distribution," in *Proceedings of the International Conference on Machine Learning (ICML)*. ACM, 2005, pp. 545–552.

[61] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing Semantic Coherence in Topic Models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2011, pp. 262–272.

[62] Y. W. Teh, "Dirichlet Process," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 280–287.

[63] S. Goldwater, T. L. Griffiths, and M. Johnson, "Producing Power-law Distribu-

tions and Damping Word Frequencies with Two-stage Language Models," *The Journal of Machine Learning Research (JMLR)*, vol. 12, pp. 2335–2382, 2011.

[64] Y. W. Teh, "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.* Association for Computational Linguistics, 2006, pp. 985–992.

[65] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[66] M. Steyvers and T. Griffiths, "Probabilistic Topic Models," *Handbook of Latent Semantic Analysis*, vol. 427, no. 7, pp. 424–440, 2007.

[67] M. Hoffman, F. R. Bach, and D. M. Blei, "Online Learning for Latent Dirichlet Allocation," in *Advances in Neural Information Processing Systems (NIPS)* , 2010, pp. 856–864.

[68] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *The journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[69] W. K. Hastings, “Monte Carlo Sampling Methods using Markov Chains and their Applications,” *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[70] S. Brooks, “Markov Chain Monte Carlo Method and its Application,” *Journal of the Royal Statistical Society: Series D (the Statistician)*, vol. 47, no. 1, pp. 69–100, 1998.

[71] L. Yao, D. Mimno, and A. McCallum, “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.* ACM, 2009, pp. 937–946.

[72] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li, “RCV1: A New Benchmark Collection for Text Categorization Research,” *The Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.

[73] P. R. Clarkson and A. J. Robinson, “Language Model Adaptation using Mixtures and an Exponentially Decaying Cache,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. IEEE, 1997, pp. 799–802.

[74] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The Kaldi Speech Recognition Roolkit,” in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[75] B. Wheatley, “Callhome Spanish Transcripts,” https://catalog.ldc.upenn.edu/LDC96T17, Linguistic Data Consortium, 1996.

[76] D. Povey, L. Burget, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiát, A. Rastrow *et al.*, “Subspace Gaussian Mixture Models for Speech Recognition,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2010, pp. 4330–4333.

[77] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, “Sequence-discriminative Training of Deep Neural Networks,” in *Proc. of Interspeech*, 2013.

[78] D. Povey *et al.*, “Generating exact lattices in the wfst framework,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.

[79] D. Miller *et al.*, “Rapid and Accurate Spoken Term Detection,” in *Proc. of Interspeech*, 2007.

[80] M. Mohri, F. Pereira, and M. Riley, “Weighted Finite-state transducers in Speech Recognition,” *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[81] J. R. Bellegarda, “Statistical Language Model Adaptation: Review and Perspectives,” *Speech Communication*, vol. 42, no. 1, pp. 93–108, 2004.

[82] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative Language Modeling with Conditional Random Fields and the Perceptron Algorithm," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 47.

[83] B.-J. Hsu, "Generalized Linear Interpolation of Language Models," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2007, pp. 136–140.

[84] C. Allauzen, M. Mohri, and B. Roark, "Generalized Algorithms for Constructing Statistical Language Models," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 40–47.

[85] J. Wintrode, "Can you repeat that? Using Word Repetition to Improve Spoken Term Detection," in *Proc. of ACL*, 2014.

[86] M. Saraclar and R. W. Sproat, "Lattice-Based Search For Spoken Utterance Retrieval," in *HLT-NAACL*, 2004.

[87] A. K. McCallum, "MALLET: A Machine Learning for Language Toolkit," 2002, `http://mallet.cs.umass.edu`.

[88] A. Stolcke, "SRILM – An Extensible Language Modeling Toolkit," in *Proc. of the International Conference on Spoken Language Processing*, 2002.

[89] L. F. Korsos and M. Taddy, “Gibbs Sampling for n-Gram Latent Dirichlet Allocation,” http://home.uchicago.edu/~lkorsos/GibbsNGramLDA.pdf, 2011, [Online; accessed Sep-2015].

[90] X. Wang, A. McCallum, and X. Wei, “Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval,” in *Proc. of IEEE International Conference on Data Mining.* IEEE, 2007, pp. 697–702.

[91] K. Zhai, J. Boyd-Graber, N. Asadi, and M. L. Alkhouja, “Mr. LDA: A Flexible Large Scale Topic Modeling Package using Variational Inference in MapReduce,” in *Proceedings of the 21st international conference on World Wide Web.* ACM, 2012, pp. 879–888.

[92] D. G. et al., “Fisher Spanish - Transcripts,” `http://catalog.ldc.upenn.edu/LDC2010T04`, Linguistic Data Consortium, 2010.

[93] H. M. Wallach, D. M. Mimno, and A. McCallum, “Rethinking LDA: Why Priors Matter,” in *Neural Information Processing Systems (NIPS)*, vol. 22, 2009, pp. 1973–1981.

[94] E. Mengusoglu and O. Deroo, “Turkish LVCSR: Database Preparation and Language Modeling for an Aglutinative Language,” in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 6. IEEE; 1999, 2001, pp. 4018–4018.

[95] R. Noyer, “Vietnamese 'Morphology' and the Definition of Word,” *University of Pennsylvania Working Papers in Linguistics*, vol. 5, no. 2, p. 5, 1998.

[96] C. May *et al.*, “Topic Identification and Discovery on Text and Speech,” in *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[97] A. Rosenberg and J. Hirschberg, “V-Measure: A Conditional Entropy-based External Cluster Evaluation Measure,” in *Proc. of the 2007 EMNLP-CoNLL Joint Conference*, 2007.

[98] G. Karypis, “CLUTO - a Clustering Toolkit,” DTIC Document, Tech. Rep., 2002.

[99] NIST, “OpenKWS13 Evaluation,” `http://www.nist.gov/itl/iad/mig/openkws13.cfm`, 2013, [Online; accessed Sep-2015].

[100] ——, “OpenKWS14 Evaluation,” `http://www.nist.gov/itl/iad/mig/openkws14.cfm`, 2014, [Online; accessed Sep-2015].

[101] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, “OpenFst: A General and Efficient Weighted Finite-State Transducer Library,” in *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, ser. Lecture Notes in Computer Science, vol. 4783. Springer, 2007, pp. 11–23, `http://www.openfst.org`.

[102] D. Caseiro and I. Trancoso, “Using Dynamic WFST Composition for Recognizing Broadcast News,” in *Proc. of Interspeech.* Citeseer, 2002.

[103] C. Allauzen, M. Riley, and J. Schalkwyk, “A Generalized Composition Algorithm for Weighted Finite-state Transducers,” in *Proc. of Interspeech*, 2009, pp. 1203–1206.

[104] J. Wintrode, G. Sell, A. Jansen, M. Fox, G.-R. Daniel, and A. McCree, “Content-Based Recommender Systems for Spoken Documents,” in *Proc. of IEEE International Confernce on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[105] B.-H. Juang and S. Katagiri, “Discriminative learning for minimum error classification,” *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 3043–3054, 1992.

Vita

Jonathan Wintrode received the A. B. degree *cum laude* in Computer Science from Harvard University in 2000, the M. S. degree in Computer Science from the Naval Postgraduate School in 2005, enrolled in the Computer Science Ph.D. program at Johns Hopkins University in 2010, and completed the M. S. E. degree in Computer Science in 2014. He won the Naval Postgraduate School Computer Science Department's Outstanding Department of Defense Student Award in 2005.