Hierarchical Bayesian Language Models for Multiparty Conversational Speech Recognition

Songfang Huang

Doctor of Philosophy

Institute for Communicating and Collaborative Systems

School of Informatics

University of Edinburgh

2010

Abstract

The goal of this thesis is to exploit rich multimodal cues available in multiparty conversational meetings to augment a traditional n-gram language model, which in turn improves the performance of automatic speech recognition for meetings. We use hierarchical Bayesian models for this task. In addition to lexical information, the multimodal cues we consider in this thesis include semantic context, participant role, and prosodic features.

We first present the application of hierarchical Pitman-Yor process language models (HPYLM) on a large vocabulary meeting ASR system. The HPYLM provides an alternative interpretation to language models in theory, and a better smoothing algorithm for language modelling in practice. We propose a parallel training algorithm to enable the training and testing the HPYLM using very large corpora. An approximate to the HPYLM, called power law discounting LM, is proposed to take advantage of power law characteristic of the HPYLM, while significantly reducing the computational requirements. Next, we introduce the modelling of topic and role information over sequential data in meetings for language modelling using the hierarchical Dirichlet process (HDP). We demonstrate that the HDP enables the inclusion of additional information for language modelling by flexibly introducing additional variables into the hierarchical Bayesian framework; and the ability of modelling the interaction between lexical information and other non-lexical cues for language models such as topic and role. Finally, we investigate the use of continuous prosodic features for language modelling via latent Dirichlet allocation (LDA). By using these hierarchical Bayesian models, we extend the traditional n-gram LMs to go beyond maximum likelihood estimation, and beyond lexical information to accommodate rich multimodal cues. We experimentally show perplexity and word error rate results on the AMI Meeting Corpus and the ICSI Meeting Corpus.

Acknowledgements

First of all, I would like to express my deepest appreciation to my supervisor Steve Renals, for his deep insights and dedication to guide and help me throughout this thesis research. Without his creative and valuable supervision, this thesis would not be possible.

Many thanks to my second supervisor Simon King for his advices and encouragement during my PhD study. I thank Sharon Goldwater for serving as the committee member and giving me valuable feedback on the earlier draft. I am extremely grateful to Miles Osborne, and Dietrich Klakow from Saarland University, for serving as examiners for my thesis.

I would like to thank the AMI-ASR team, for providing the baseline ASR system for my experiments, especially Thomas Hain, Vincent Wan, Guilia Garau, and Mike Lincoln for helpful discussions.

I am extremely grateful to the Overseas Research Scheme (ORS), the Wolfson Microelectronics Scholarship, and the AMI and AMIDA projects, for their generous funding to support my PhD study. Many thanks also go to the EMIME project for supporting me to attend ICASSP'10 at Dallas.

Many thanks to Michael Picheny and Bowen Zhou at IBM, for giving me an opportunity to do an internship at IBM T. J. Watson Research, and thereafter offering me a PostDoc position. I really enjoy the work with the speech-to-speech translation team at IBM. I would also thank Pei-yun Hsueh for her kind help during my relocation to IBM.

I am grateful to my Master supervisors, Huisheng Chi and Xihong Wu, at Peking University. I enjoy and benefit from every discussion with you when I go back to China.

I would also like to thank my Chinese colleagues and friends, Le Zhang and Dong Wang from CSTR, and Xinkun Liu, Yansong Feng, Zheng Wang, from Informatics. I really enjoyed the time with you at Edinburgh. I also want to all the folks at CSTR, including my office mates: Tim Mills, Peter Bell, Parthal Lal, Leolardo Badino, and Oliver Watts. It is a great experience to work with you at CSTR.

Finally, my biggest thanks goes to my wife Yiou Zheng for her continuous love and support in past years, and my parents for their encouragement.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Songfang Huang*)

To my family.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Statistical language models (LM) are an essential component of speech and language processing for human-computer interaction, being a key component of systems for automatic speech recognition (ASR) [Jelinek, 1997], machine translation [Brown et al., 1991], and information retrieval [Ponte and Croft, 1998; Zhai and Lafferty, 2004]. A language model forms a probability distribution over sentences. The n-gram model [Jelinek, 1997] is the most widely used language modelling approach in the state-of-the-art large vocabulary speech recognition systems.

In this thesis, we are interested in hierarchical Bayesian approaches [Gelman et al., 2004] to language modelling, in the context of multiparty conversational meeting speech recognition. We demonstrate that hierarchical Bayesian models provide a coherent framework for language modelling with the following advantages, compared to traditional n-gram models:

- an alternative interpretation to language models in theory, and a better smoothing algorithm for language modelling in practice;
- the inclusion of additional information for language modelling, by flexibly introducing additional variables into the hierarchical Bayesian framework;
- the ability of modelling the interaction between lexical information and other non-lexical cues for language models.

1.1 Background

We now live in a data-intensive information age. How we can search, interpret, summarise and visualize useful information from such large quantities of data, therefore,

becomes more and more important for us. Computers have proven to be an efficient tool to help the acquisition, processing and presentation of information. In addition, computers, together with networks, play an important role in group communication and collaboration, making human interaction more robust and effective. Computer-enhanced systems and applications have become more and more popular in everyday life.

Group meetings are one example to illustrate the importance of computer assistance. Social psychologists have studied meetings for over fifty years, with the general purpose of understanding group dynamics and communication [Bales, 1951]. With a view to information access from meetings, we note that, during a meeting, there are data coming from multiple sources (i.e., simultaneous multi-streams of different meeting participants) or from multiple modalities (i.e., audio, video, text, etc.). It is hard for people to manually analyse and acquire information from these raw and unstructured data. Therefore, an automatic way to analyse *unstructured* meeting archives, to eventually provide *structured* information for people to browse and retrieval will be helpful, not only for offline access to meeting archives but also for online human interaction.

Research on computer-supported meetings has received considerable attentions, especially from the artificial intelligence (AI) and speech processing communities. Generally speaking, this emerging field focuses on automatically analysing and structuring large volumes of meeting archives to enable easily browsing, and more sophisticatedly, to enable online assistance for face-to-face or remote human communication. The AMI (Augmented Multi-party Interaction) project and its follow-up AMIDA (Augmented Multi-party Interaction with Distance Access)[1] [Renals et al., 2007] are European-funded research projects on meetings with international involvements including partners from universities, institutes and companies. Other meeting research projects include ICSI Meeting Recorder project [Morgan, 2001], CMU Meeting Record Creation and Access project [Waibel et al., 2001], and CALO(Cognitive Assistant that Learns and Organizes)[2] [Tur et al., 2010] from USA, and M4 (MultiModal Meeting Manager)[3], IM2 (Interactive Multimodal Information Management), and CHIL(Computers in the Human Interaction Loop)[4] [Waibel and Stiefelhagen, 2009] from Europe, and so on.

The Automatic transcription of meeting speech is typically one of the first of sev-

[1] http://www.amiproject.org
[2] http://caloproject.sri.com
[3] http://www.m4project.org
[4] http://chil.server.de

eral essential steps for processing of meetings. Meeting recognition is much more challenging than that for read speech, due to the multiparty and conversational characteristics of meeting speech. For example, spontaneous meeting speech is recorded from multiple simultaneous participants in various noise conditions. It is therefore common to find filler pauses, back-channels, and overlapping speech in meetings. Human interactions in meetings are multimodal [Krauss et al., 1977]. However, most ASR systems nowadays rely solely on audio, without considering the multimodal characteristics of human perception. How to efficiently incorporate multimodal cues into current ASR frameworks remains an open question in the speech research community. In addition, the most widely used n-gram models are relying on only lexical information. To this end, we in this thesis address the inclusion of multimodal cues for the automatic recognition of multiparty conversational meeting speech, via the language model component in an ASR system.

1.2 Thesis Overview

The goal of this thesis is to exploit rich multimodal cues available in multiparty conversational meetings to augment a traditional n-gram language model, which in turn improves the performance of automatic speech recognition for meetings. In general, we use hierarchical Bayesian models [Gelman et al., 2004] for this task. Figure 1.1 shows a roadmap to the hierarchical Bayesian framework we proposed in this thesis for language modelling in multiparty conversational speech recognition.

More precisely, the hierarchical Bayesian framework shown in Figure 1.1 makes clear the following three questions we will address in the rest of this thesis.

1. **How to estimate a better n-gram LM based on lexical information?** Smoothing is of central importance to estimate a better language model. There are a number of smoothing techniques in the literature [Chen and Goodman, 1999]. As illustrated at the top of Figure 1.1, we in this thesis address this question using Bayesian language models. For lexical information, we will investigate in Chapter 4 hierarchical Dirichlet language models (HDLM) [MacKay and Peto, 1994], and hierarchical Pitman-Yor process language models (HPYLM) [Teh, 2006b; Goldwater et al., 2006b] on large vocabulary speech recognition for meetings using large training corpora. The Bayesian language model based on the Pitman-Yor process has been demonstrated as not only a Bayesian interpretation to smoothing in theory [Teh, 2006b], but also a better technique for smooth-

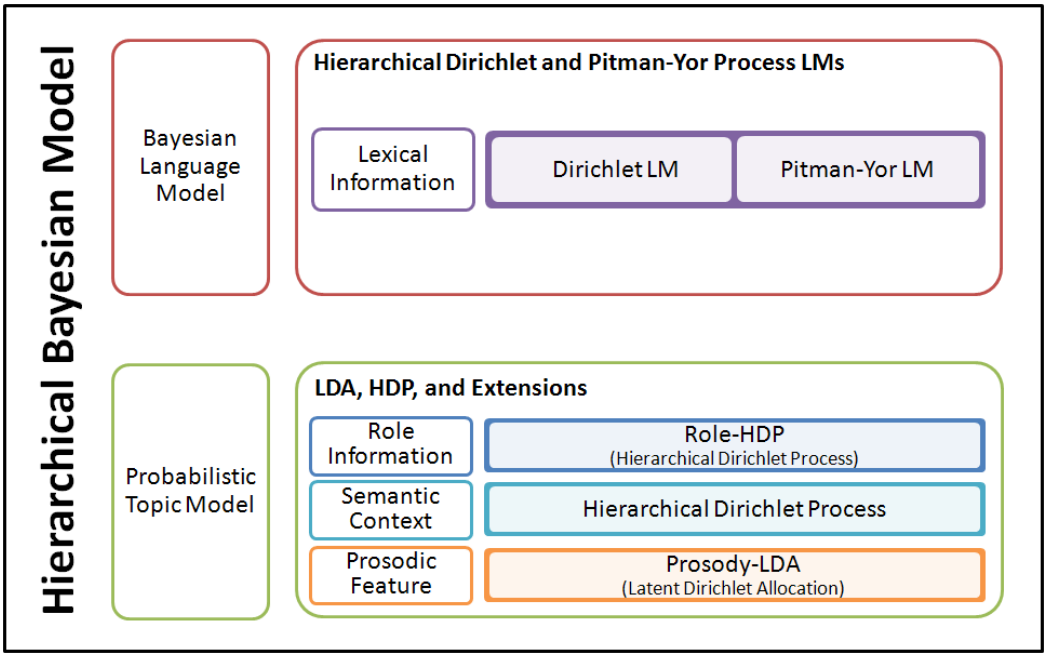


Figure 1.1: A roadmap to the hierarchical Bayesian framework for language modelling for automatic recognition of multiparty conversational meeting speech.

ing in practice than state-of-the-art methods. We also propose an approximate to the HPYLM and obtain a power law discounting LM in Chapter 5.

2. **How to extract and represent potential multimodal cues in meetings?** The multimodal cues considered in this thesis include semantic context, role of participants, and suprasegmental features, as shown at the bottom of Figure 1.1. These heterogeneous multimodal cues are at different levels (meeting, speaker, ...), at different scales (syllable, word, sentence, document, ...), and of different types (lexical, semantic, social information, audio, ...). We will review some previous research on using multimodal cues for LMs/ASR in Chapter 3. The extraction and representations for these multimodal cues could be significantly different for different cues. We address this question using various hierarchical Bayesian models in this thesis. Suprasegmental features, for example, have a temporal scale compared to other cues. It is therefore important to obtain an intermediate representation between suprasegmental features and words, which turns out to be a difficult task (we will revisit this in Chapter 7). Cues like semantic context must be inferred first, based on other information, before they can be used for language modelling. Topic models, such as latent Dirichlet allocation (LDA) [Blei et al., 2003] and hierarchical Dirichlet process (HDP) [Teh et al.,

2006], are used in Chapter 6 to infer the semantic context in an unsupervised way. The flexibility of including additional variables into a hierarchical Bayesian model makes it easy to represent and incorporate role information within the HDP model. The HDP is extended to accommodate aspects such as participant role by introducing additional level in the HDP hierarchy in Chapter 6.

3. **How to combine lexical information and multimodal cues in LMs?** One simple way to combine lexical information and multimodal cues in LMs is to train separate LMs on different modalities, and then combine them using linear interpolation. Alternatively, LMs trained on multimodal cues can be used as dynamic marginal to adapt baseline LMs trained on lexical information [Kneser et al., 1997], as we did in Chapter 6 to combine topic information with n-gram LMs. We further attempt to combine n-gram LMs and hierarchical Dirichlet process models in Chapter 6, which is an extension to the "beyond bag-of-word" model in [Wallach, 2006]. We also apply LDA topic models to incorporate continuous suprasegmental prosodic features with discrete lexical information for language modelling in Chapter 7. As illustrated in Figure 1.1, however, the ideal way for the combination is supposed to be done within a hierarchical Bayesian framework, by combining a Bayesian language model and an augmented probabilistic topic model via coherent Bayesian inference.

In summary, this thesis concentrates on the incorporation of multimodal cues, ranging from lexical information, to semantic topics, social roles, and prosodic speaking styles, into LMs using hierarchical Bayesian models, for *multiparty* and *conversational* speech recognition in meetings.

1.3 Thesis Contributions

This thesis contributes to the language modelling for multiparty conversational speech in the following ways:

- We carry out a comprehensive investigation on Bayesian language models for large vocabulary continuous speech recognition for meetings. Previous work mainly focuses on applications on text processing and evaluates in perplexity. We instead verify the effectiveness of Bayesian language models on practical ASR systems in terms of both perplexity and word error rate (WER).

- We propose a parallel training algorithm for Bayesian language models, which enables the application of Bayesian language models on large scale ASR tasks using large training corpora.
- We provide an approximation to Bayesian language models, and obtain a power law discounting language model (PLDLM). The PLDLM retains one remarkable behaviour of the HPYLM – the power law discounting, while significantly reduces the computational requirements for training an PLDLM.
- We develop various hierarchical Bayesian models to accommodate multimodal cues in meetings, i.e., semantic context, participant role, and prosodic features, and incorporate multimodal cues in different ways to enrich a lexical-based LM.
- Finally, we investigate the scalability of these hierarchical Bayesian models on large scale ASR tasks for multiparty conversational speech.

1.4 Thesis Organization

The remainder of this thesis is structured as follows:

We begin with a literature review on statistical models for speech and language in the next chapter. We briefly introduce the general framework for an automatic speech recognition system. The statistical language model, one of the main component in an ASR system, is described, covering the maximum likelihood estimation (MLE), smoothing, and extensions. We review hierarchical Bayesian models, by first introducing some common Bayesian priors and Bayesian inference methods, and then giving an overview on Bayesian language modelling and probabilistic topic modelling.

Chapter 3 provides a literature review on research on multiparty conversational meetings. We focus on the meeting transcription task. After summarising some research projects on meetings, we introduce two meeting corpora, and then specifically review some multimodal cues available in meetings, and their uses for language modelling according to previous research. The AMI-ASR baseline system for meetings will also be described in Chapter 3.

There are four main chapters in this thesis, beginning from Chapter 4, which introduces our work on the application of hierarchical Pitman-Yor process language models [Teh, 2006b; Goldwater et al., 2006b]. The main contributions of this work are two-fold: firstly we make the HPYLM scalable to large vocabulary speech recognition using large text corpora, via an efficient implementation and parallel computing

for the HPYLM; and secondly we experimentally demonstrate the effectiveness of the HPYLM on a practical meeting speech recognition task, in terms of both perplexity and word error rate (WER).

We introduce the power law discounting LM, an approximation to the HPYLM, in Chapter 5. The PLDLM introduces more free parameters, which are directly estimated using a power law form, for language model smoothing. The PLDLM preserves the power low discounting characteristic in the HPYLM, and improves upon the state-of-the-art smoothing methods for ASR.

Chapter 6 introduces an application of hierarchical Dirichlet process [Teh et al., 2006] to model semantic topics, and its extension — roleHDP — to accommodate the role of participant information. Topics extracted by the HDP serve as abstract representations for latent structure. The roleHDP additionally relates the topics and roles, and is used in Chapter 6 for role prediction. We use the semantic topics and role information for language modelling adaptation, based on the dynamic unigram marginal method [Kneser et al., 1997]. Experimental results on the AMI Meeting Corpus show significant reduction in word error rate. In Chapter 6, we also attempt to combine a traditional n-gram LM with the hierarchical Dirichlet process model [Teh et al., 2006] within a coherent Bayesian framework, and obtain the hierarchical Dirichlet process LM (HDPLM).

Chapter 7 exploits a different type of multimodal cue – prosodic feature from conversational speech. It is continuous, suprasegmental, and noisy. We focus on the extraction of prosodic features at the unit of syllable, and the representations that are compatible with lexical word for prosodic features. Approaches based on hierarchical Bayesian models show better performance than models based on maximum likelihood estimation.

In the last chapter, we summarise the main contributions of this thesis, and discuss some potentially interesting future work.

1.5 Published Work

Some of the work presented in this thesis has been published. Chapter 4 covers [Huang and Renals, 2010b], which is an extension and combination of [Huang and Renals, 2007a] and [Huang and Renals, 2009]. The work in Chapter 5 on the power law discounting LM was published in [Huang and Renals, 2010a].

Part of Chapter 6 was published in [Huang and Renals, 2008b,c,a].

The work on using prosody for language modelling in Chapter 7 was published in [Huang and Renals, 2007b].

Chapter 2

Statistical Models for Speech and Language

Statistical models are currently the mainstream approaches to many applications, such as automatic speech recognition and natural language processing. A statistical model is a set of probability distributions on the sample space Θ, with a parameter set θ together with a function $P : \theta \rightarrow \mathcal{P}(\Theta)$ which assigns to each parameter point $\theta_i \in \theta$ a probability distribution P_{θ_i} on Θ [McCullagh, 2002]. Here $\mathcal{P}(\Theta)$ is the set of all probability distribution on Θ. In this chapter, we will introduce and give a literature review on some relevant statistical models, with focus on statistical language models and hierarchical Bayesian models. A Bayesian model requires an additional prior distribution on θ.

2.1 Automatic Speech Recognition

The main goal of this thesis is to investigate new approaches to estimate an augmented language model to be used in the task of automatic recognition of conversational speech in multiparty meetings. The role of a language model within the automatic speech recognition (ASR) framework is to provide a constrained search space of linguistic knowledge, to search for the most probable word sequence $\hat{\mathbf{W}}$ corresponding to an acoustic signal $\mathbf{A}$:

$$\hat{\mathbf{W}} = \arg\max_{w} P(\mathbf{W}|\mathbf{A}) = \frac{P(\mathbf{A}|\mathbf{W}) \cdot P(\mathbf{W})}{P(\mathbf{A})} \tag{2.1}$$

where the term $P(\mathbf{A}|\mathbf{W})$, usually referred to as the *acoustic model*, represents the likelihood that the acoustic signal $\mathbf{A}$ can be generated from the word sequence $\mathbf{W}$, and

$P(\mathbf{W})$ is the probability of the word sequence $\mathbf{W}$ and is often called a *language model*.

Automatic speech recognition converts an input speech signal into a sequence of most probable words. Figure 2.1 shows the core components in a typical ASR system.

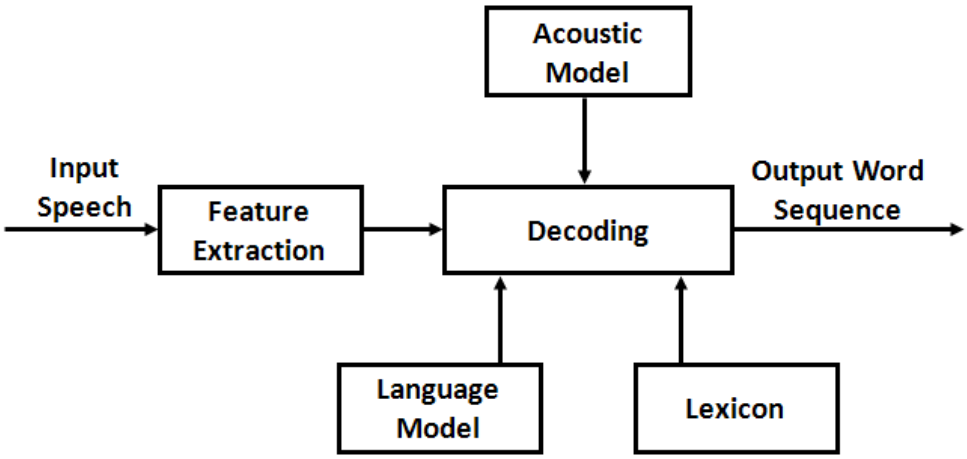


Figure 2.1: The components of a typical automatic speech recognition system.

- **Feature Extraction**: the speech signal is parameterised into sequence of frames over time, each of which is represented as a low-dimensional vector. Popular feature representations include Mel frequency cepstral coefficients (MFCC) [Davis and Mermelstein, 1980] and perceptual linear prediction (PLP) coefficients [Hermansky, 1990]. In a typical ASR system, dynamic features, i.e., the first order (delta) and second-order (delta-delta) temporal derivatives of "static" MFCC/PLP features, are heuristically appended to explicitly model the dynamics of speech signal at each time instance [Furui, 1986].

- **Acoustic Modelling**: the hidden Markov model (HMM) has been the dominant statistical model for acoustic modelling $P(\mathbf{A}|\mathbf{W})$ in ASR for decades [Rabiner, 1989]. There are two stochastic processes in an HMM: a hidden Markov chain, which accounts for temporal variability, and an observable process, which accounts for spectral variability. Each word w_i (or subword unit such as phoneme) is modelled by an HMM. Each HMM state will have a different output distribution, which typically is a mixture of Gaussians, for each observed low-dimensional feature vector a_i. An HMM for a sequence of words or phonemes can be formed by concatenating the individual trained hidden Markov models for the separate words and phonemes. The parameters of an HMM can be efficiently estimated from a training corpus using expectation-maximization (EM) algorithm [Rabiner, 1989].

- **Language Modelling**: the language model (LM) $P(\mathbf{W})$ estimates the probability of a word sequence $\mathbf{W} = (w_1, \ldots, w_M)$. We will discuss language modelling in details in the rest of this thesis.

- **Lexicon**: the lexicon provides pronunciations for words, by mapping words into smaller linguistic units suitable for use in HMM-based acoustic models, such as phonemes or graphemes. Those words that do not occur in a lexicon are called out-of-vocabulary (OOV) words. How to deal with OOV words in an ASR system remains an open question.

- **Decoding**: decoding finds the best word sequence $\hat{\mathbf{W}}$ by considering the information from both acoustic and language models according to Equation (2.1). This corresponds to finding the most likely state sequence in the network of HMM states. The Viterbi algorithm, a dynamic programming algorithm, is mostly used for decoding in state-of-the-art ASR systems.

Apart from automatic speech recognition, language models have been widely studied within other domains, such as machine translation, information retrieval, and optical character recognition.

2.2 Statistical Language Model

The task of a language model is to provide a probability distribution over a string $W = (w_1, \ldots, w_M)$, where the individual elements w_i are drawn from a set of word types called vocabulary, V. The joint probability distribution $P(w_1, \ldots, w_M)$ over words can be decomposed into the multiplication of a sequence of conditional distributions, as expressed in Equation (2.2).

$$\begin{aligned} P(w_1, \ldots, w_M) &= P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\ldots P(w_M|w_1, \ldots, w_{M-1}) \\ &= \prod_{i=1}^{M} P(w_i|w_1, \ldots, w_{i-1}) \end{aligned} \tag{2.2}$$

where $\mathbf{u}_i = w_1^{i-1} = w_1, \ldots, w_{i-1}$ is referred to as the *history* or *context* for word w_i. In this sense, the goal of statistical language modelling can also be described as estimating the probabilities for the next word conditioned on the sequence seen so far, i.e., $P(w_i|w_1, \ldots, w_{i-1})$.

Although longer contexts would be more desirable for predicting the next word, finite contexts, i.e., the immediately preceding $n-1$ words for some fixed n, w_{i-n+1}^{i-1},

are used to compute the conditional probabilities for w_i instead of full context w_1^{i-1}, for the sake of mathematical and computational simplicity. The resulting model, named n-gram, is the most widely-used language model for speech and language processing. In this way, Equation (2.2) can be approximated by Equation (2.3) according to the Markov chain assumption:

$$\begin{aligned} P(w_1,\ldots,w_M) &= \prod_{i=1}^{M} P(w_i|w_1,\ldots,w_{i-1}) \\ &\approx \prod_{i=1}^{M} P(w_i|w_{i-n+1},\ldots,w_{i-1}) \end{aligned} \tag{2.3}$$

For the case of $n = 2$, the n-gram models are called *bigram* models, and $n = 3$ *trigram* models.

Significant progress and extensions have been made to language models during recent decades. However, many of those extensions can be more broadly regarded as applying some different mapping functions $\Phi(\mathbf{u}_i)$ over the context $\mathbf{u}_i$. In this way, Equation (2.2) leads to a more general form: $P(W) = P(w_1,\ldots,w_M) = \prod_{i=1}^{M} P(w_i|\Phi(\mathbf{u}_i))$.

Language models are important components in automatic speech recognition systems, where they probabilistically constrain the language that the system could recognize, and form the search space for decoding. With the use of LMs in ASR, more probable sentences will be distinguished from those that are acoustically similar. This is important and helpful for the cases where speech signals are presented with noise.

The most commonly used metric for quantitatively measuring the performance of LMs is *perplexity*, which can be thought of the number of words which must be selected from in the predictive distributions of LMs. Perplexity relates to cross-entropy ($H(T)$ in Equation (2.4)), and is equal to the geometric average of the inverse probabilities of the words measured on the test data T:

$$PPL(T) = 2^{H(T)} = 2^{-\frac{1}{M}\log_2 \prod_{i=1}^{M} p(w_i|\mathbf{u}_i)} = \sqrt[M]{\prod_{i=1}^{M} \frac{1}{P(w_i|\mathbf{u}_i)}} \tag{2.4}$$

where M is the total number of words in T.

There are also indirect methods to evaluate the performance of LMs, for example, plugging specific LMs into real applications such as ASR and statistical machine translation (SMT), and evaluating the performance in terms of word error rate (WER) for ASR and Bilingual Evaluation Understudy (BLEU) [Papineni et al., 2002] for SMT.

2.2.1 Maximum Likelihood Estimation and Smoothing

As indicated by Equation (2.3), the goal of a language model is to estimate the conditional probability distribution $P(w_i|\mathbf{u}_i)$ of the next word w_i given the context $\mathbf{u}_i$, i.e., the $n-1$ previously seen words w_{i-n+1}^{i-1}. One simple approach to this estimation is based on the maximum likelihood principle, leading to maximum likelihood estimation (MLE) for language models. MLE interprets the conditional probability distribution of LMs as a set of discrete distributions over words (one for each context), and treats these discrete distributions as being independent of each other. According to this independency assumption, the MLE for predictive distributions $P(w|\mathbf{u})$ of LMs is then given by

$$P(w|\mathbf{u}) = \arg\max_{P} \prod_{i=1}^{M} P(w_i|\mathbf{u}_i) \tag{2.5}$$

Solving the optimization in the Equation (2.5), the maximum likelihood estimation for $P(w_i|\mathbf{u}_i)$ is obtained from training data as a relative frequency

$$P(w_i|\mathbf{u}_i) = \frac{C(\mathbf{u}_i w_i)}{\sum_{i'} C(\mathbf{u}_i w_{i'})} = \frac{C(\mathbf{u}_i w_i))}{C(\mathbf{u}_i)} = \frac{C(w_{i-n+1}^{i-1} w_i)}{C(w_{i-n+1}^{i-1})} \tag{2.6}$$

where $C(\cdot)$ denotes the count of all occurrences of $\cdot$ in training data.

There are several problems arising from maximum likelihood estimation of LMs. The first problem is caused by the *curse of dimensionality*. Simply consider the case of $n = 3$ and a vocabulary with 50000 words, which requires 50000^3 parameters to be estimated. As a result, MLE will assign zero probability to words that never occur corresponding to certain contexts in training data. This will be problematic in some applications, i.e., speech recognition where the recognized word sequence $\hat{W}$ is found by the maximizing $P(W|A) \propto \arg\max_W P(A|W)P(W)$. If the probability of language model $P(W)$ were assigned zero, the string W will never be considered as a potential transcription, regardless of how unambiguous the acoustic signal A is [Chen and Goodman, 1999]. To avoid this problem, it therefore requires the language model $P(W)$ to assign a non-zero probability for a potential hypothesis during decoding in ASR. Another problem of MLE for LMs is caused by data sparsity. Even with a relatively large amount of training data, it is inevitable that the vast majority of word combinations occur very infrequently, making it difficult to estimate accurate probabilities of words in most contexts. Finally, LMs that are estimated by maximizing the likelihoods of a particular training data suffer seriously from the poor generalisation of LMs to other test data. Direct MLE therefore will make resulting LMs severely overfitted to the training data.

Smoothing may be used to alleviate these problems. The common sense for smoothing is that there are some correlations or regularities among contexts $\mathbf{u}$ for predictive distribution $P(w_i|\mathbf{u})$. The regularities can intuitively be made use of to share knowledge between those seen and unseen word strings. In general, there are a number of smoothing approaches in LMs [Chen and Goodman, 1999], for example, *discounting*, *back-off*, *interpolation*, and a combination of these smoothing techniques.

2.2.1.1 Discounting

Discounting redistributes the probability densities, by adjusting low probabilities upward, and high probabilities downward, to estimate a more uniform probability distribution for LMs. It is normally used in combination with back-off or interpolation. The probability mass of observed events is discounted, such that probabilities for unseen events in training data can be assigned from the leftover probability mass. Typical discounting algorithms include Good-Turing discounting [Good, 1953], Witten-Bell discounting [Witten and Bell, 1991], Absolute discounting [Ney et al., 1994], and Kneser-Ney discounting [Kneser and Ney, 1995].

Good-Turing smoothing [Good, 1953; Katz, 1987], which is based on leave-one-out estimation, re-estimates the probability mass assigned to n-grams with zero counts by making use of the frequency of n-grams occurring only once. In this way, Good-Turing smoothing provides a simple way to estimate the total probability for unseen n-grams, which is subsequently divided and uniformly redistributed among those unseen n-grams. For any n-gram that occurs r times in the Good-Turing estimate, we pretend that it occurs r^* times instead, where

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{2.7}$$

and where n_r is the number of n-grams that occur exactly r times in the training data. For example, we assign each n-gram with a zero count with a count of n_1/n_0. The smoothed probability for an n-gram $\mathbf{u}_i w_i$ with r counts is obtained by normalizing the count in Equation (2.7), we take

$$P(w_i|\mathbf{u}_i) = \frac{r^*}{N} \tag{2.8}$$

where $N = \sum_{r=0}^{\infty} n_r r^*$.The Good-Turing method is of central importance to many other smoothing techniques, and it is normally not used by itself for n-gram smoothing, but used in combination with other smoothing techniques such as back-off and interpolation. Katz smoothing [Katz, 1987], for example, combines the Good-Turing estimate and the interpolation of higher order models with lower order models.

Absolute discounting [Ney et al., 1994], the root of Kneser-Ney smoothing [Kneser and Ney, 1995], subtracts a fixed absolute discount d from each non-zero count, and redistributes the unallocated probability mass to those unseen events. The intuition is due to two reasons [Jurafsky and Martin, 2009]: firstly, we have good estimates already for n-grams with high counts, and a small discount d won't affect them much; secondly, absolute discounting will mainly modify those small counts for which we don't necessarily trust the estimate anyway. The equation for absolute discounting in combination with back-off is shown in Equation (2.9). Absolute discounting can also be used in interpolated LMs.

$$P_{\text{abs}}(w_i|\mathbf{u}_i) = \begin{cases} \frac{C(\mathbf{u}_i w_i)-d}{C(\mathbf{u}_i)} & \text{if } C(\mathbf{u}_i w_i) > 0 \\ \alpha(\mathbf{u}_i) P_{\text{abs}}(w_i|\pi(\mathbf{u}_i)) & \text{otherwise} \end{cases} \tag{2.9}$$

where the discount d ranges from 0 to 1, and $\alpha(\mathbf{u}_i)$ are the normalization factors to make sure that $\sum_w P_{\text{abs}}(w|\mathbf{u}_i) = 1$

In contrast to discounting schemes that slice off counts and redistribute, there are also some discounting schemes that add pseudocounts. Additive smoothing, or add-δ smoothing [Laplace, 1825; Lidstone, 1920; Johnson, 1932; Jeffreys, 1948] is one of the simplest smoothing methods, which adds a pseudocount δ to every count where typically $0 \leq \delta \leq 1$, as shown in Equation (2.10).

$$P_{\text{add}}(w_i|\mathbf{u}_i) = \frac{C(\mathbf{u}_i w_i) + \delta}{C(\mathbf{u}_i) + \delta|V|} \tag{2.10}$$

where V is the vocabulary. If $\delta = 1$, this is also called Laplace smoothing [Lidstone, 1920; Jeffreys, 1948]. Additive smoothing, however, generally performs poorly in practice.

2.2.1.2 Back-off

It is also useful to take advantage of the n-gram hierarchy for smoothing methods through back-off [Katz, 1987] or interpolation [Jelinek and Mercer, 1980]. Back-off, first introduced by [Katz, 1987] and hence also called Katz back-off, utilizes the correlations between context (i.e., $\mathbf{u}_i = w_{i-n+1}^{i-1}$) of an nth order LM and that (i.e., $\pi(\mathbf{u}_i) = w_{i-n+2}^{i-1}$) of an $(n-1)$th order LM where $\pi(\mathbf{u}_i)$ denotes the back-off context for $\mathbf{u}_i$, i.e., the context that is one word shorter than $\mathbf{u}_i$. In a Katz back-off n-gram model, an n-gram with a non-zero counts is calculated as a discounted probability, while an n-gram with a zero count is approximated by backing off to the $(n-1)$-gram, as shown

in Equation (2.11).

$$P_{\text{Katz}}(w_i|\mathbf{u}_i) = \begin{cases} P^*(w_i|\mathbf{u}_i) & \text{if } C(\mathbf{u}_i w_i) > 0 \\ \alpha(\mathbf{u}_i) P_{\text{katz}}(w_i|\pi(\mathbf{u}_i)) & \text{otherwise} \end{cases} \tag{2.11}$$

This back-off process iterates until the unigram, i.e., back-off from a trigram to bigram and eventually unigram. In this way, the problem of assigning zero probabilities can be avoided, because the lower the order is, the more reliable the counts will be.

The discounted probability $P^*(w_i|\mathbf{u}_i)$ is calculated using the discounted count C^* predicted by the Good-Turing estimate in Equation (2.7), that is

$$P^*(w_i|\mathbf{u}_i) = \frac{C^*(\mathbf{u}_i w_i)}{C(\mathbf{u}_i)} \tag{2.12}$$

Some probability mass will be left over for the lower order n-grams and redistributed by the α weights. In practice, larger counts are considered to be more reliable, so they are not discounted in Katz back-off n-gram LMs. Katz took $C^*(\mathbf{u}_i w_i) = C_{\text{MLE}}(\mathbf{u}_i w_i)$ for all $C_{\text{MLE}}(\mathbf{u}_i w_i) > k$ for some k, where [Katz, 1987] suggested $k = 5$.

The back-off weights $\alpha(\mathbf{u}_i)$ are used to make sure that $P_{\text{Katz}}(w_i|\mathbf{u}_i)$ in Equation (2.11) satisfies the constraint of $\sum_w P_{\text{Katz}}(w|\mathbf{u}_i) = 1$. The solution to this constraint is

$$\alpha(\mathbf{u}_i) = \frac{1 - \sum_{w_i : C(\mathbf{u}_i w_i) > 0} P^*(w_i|\mathbf{u}_i)}{1 - \sum_{w_i : C(\pi(\mathbf{u}_i) w_i) > 0} P^*(w_i|\pi(\mathbf{u}_i))} \tag{2.13}$$

where the numerator represents the leftover probability mass to the lower order n-grams, and the denominator is the total probability of all the $(n-1)$-grams that have zero counts.

2.2.1.3 Interpolation

Instead of relying on only the subsequent lower order LMs in back-off smoothing, interpolation, also known as Jelinek-Mercer smoothing [Jelinek and Mercer, 1980] alternatively combines all the lower order models in a weighted form,

$$P_{\text{interp}}(w_i|\mathbf{u}_n) = \lambda_{\mathbf{u}_n} \hat{P}(w_i|\mathbf{u}_n) + \lambda_{\mathbf{u}_{n-1}} \hat{P}(w_i|\mathbf{u}_{n-1}) + \ldots + \lambda_{\mathbf{u}_0} \hat{P}(w_i|\mathbf{u}_0) \tag{2.14}$$

where $\hat{P}(w_i|\mathbf{u}_n)$ is the n-order back-off model with the context $\mathbf{u}_n$ of length $n-1$, and $\lambda_{\mathbf{u}_n}$ are mixing weights with the constraint that $\sum_n \lambda_{\mathbf{u}_n} = 1$. The mixing weights $\lambda_{\mathbf{u}_n}$ can be estimated by cross-validation on held-out data (deleted interpolation), using the Baum-Welch algorithm [Baum, 1972]. Bahl et al. [1983] further suggested partitioning the $\lambda_{\mathbf{u}_n}$ into buckets according to context count $C(\mathbf{u}_n)$, and all $\lambda_{\mathbf{u}_n}$ in the same bucket share the same value.

2.2.1.4 Advanced Smoothing Methods

To obtain a smoother LM, interpolated Kneser-Ney smoothing [Kneser and Ney, 1995] utilizes absolute discounting, modified counts for lower order m-gram probabilities ($m < n$), and interpolation with low order m-gram probabilities:

$$P_{\text{KN}}(w|\mathbf{u}) \approx P_{\mathbf{u}}^{\text{IKN}}(w) = \frac{\max(c_{\mathbf{u}w} - d_{|\mathbf{u}|}, 0)}{c_{\mathbf{u}\bullet}} + \frac{d_{|\mathbf{u}|} t_{\mathbf{u}\bullet}}{c_{\mathbf{u}\bullet}} P_{\pi(\mathbf{u})}^{\text{IKN}}(w) \tag{2.15}$$

where $c_{\mathbf{u}\bullet} = \sum_{w'} c_{\mathbf{u}w'}$ is the total number of word tokens following the context $\mathbf{u}$, $\pi(\mathbf{u})$ is the context one word shorter than $\mathbf{u}$, $t_{\mathbf{u}\bullet} = |\{w' : c_{\mathbf{u}w'} > 0\}|$ is the number of distinct words w' occurring after $\mathbf{u}$, and the discount $d_{|\mathbf{u}|}$ is dependent on the length of the context. The Kneser-Ney smoothing was motivated the *marginal constraints*, by selecting the lower order probability distribution such that the marginals of the higher order smoothed distribution (the left-hand side of Equation (2.16)) match the marginals of the training data (the right-hand side of Equation (2.16)) [Kneser and Ney, 1995]

$$\sum_{w' : w'\pi(\mathbf{u}) = \mathbf{u}} P_{\text{KN}}(w|\mathbf{u}) = \frac{C(\pi(\mathbf{u})w)}{C(\pi(\mathbf{u}))} \tag{2.16}$$

By solving the marginal constraints in Equation (2.16), [Kneser and Ney, 1995] derived modified back-off distribution for lower order n-grams according to the number of different word *types* that follow each context. The use of a modified back-distribution is the core and the largest influence for Kneser-Ney smoothing [Chen and Goodman, 1999].

Modified Kneser-Ney smoothing extends interpolated Kneser-Ney smoothing by allowing three different discount parameters, $d_{|\mathbf{u}|1}$, $d_{|\mathbf{u}|2}$, and $d_{|\mathbf{u}|3+}$ for n-grams with one, two, and three or more counts respectively [Chen and Goodman, 1999]. Counts of counts statistics are used to estimate the optimal values for average discounts in the modified Kneser-Ney smoothing, as shown in Equation (2.17).

$$\begin{aligned} d_{|\mathbf{u}|1} &= 1 - 2Y\frac{n_2}{n_1} \\ d_{|\mathbf{u}|2} &= 2 - 3Y\frac{n_3}{n_2} \\ d_{|\mathbf{u}|3+} &= 3 - 4Y\frac{n_4}{n_3} \end{aligned} \tag{2.17}$$

where $Y = \frac{n_1}{n_1 + 2n_2}$, and n_r is the number of n-grams that occur exactly r times in the training data.

Interpolated and modified Kneser-Ney smoothing are the state-of-the-art smoothing techniques for language modelling. As an alternative to these smoothing schemes,

class-based n-gram language models [Brown et al., 1992] have been used, in which word classes or clusters (either hand-designed or automatically induced) help to address the data sparsity problem. Classes for words can be manually designed, or automatically induced from corpora according to the similarities of meanings, syntax, and semantics. One word from the vocabulary can belong to one class (hard-clustering), or may belong to multiple classes (soft-clustering) with probability estimated using an EM algorithm [Saul and Pereira, 1997].

The predictive probability in a class-based bigram LM is

$$P(w_i|w_{i-1}) \approx P(c_i|c_{i-1}) \cdot P(w_i|c_i) \tag{2.18}$$

where c_i is the class of word w_i, and parameters can be learned by MLE from training data, i.e., $P(w_i|c_i) = \frac{C(w_i)}{C(c_i)}$ and $P(c_i|c_{i-1}) = \frac{C(c_{i-1}c_i)}{\sum_{c'} C(c_{i-1}c')}$. Standard bigram models are a special case of class-based bigram models in which each word is mapped to a unique word class. Normally the number of classes is much smaller than that of word vocabularies. In this sense, class-based n-grams can also be considered as a way of reducing the dimensionality of contexts, which is useful for dealing with sparsity in training data for LMs. An alternative method of dimensionality reduction for contexts, which uses distributed representations for contexts, will be introduced in Section 2.2.2.1.

Since [Brown et al., 1992], there are a number of research has been done to improve class-based language modelling, such as [Kneser and Ney, 1993; Bellegarda and Butzberger, 1996; Martin et al., 1998; Gao et al., 2002; Emami and Jelinek, 2005]. These are mainly working on better word clustering algorithms for class-based language models.

Based on the empirical comparisons of smoothing techniques for language modelling, the conclusions drew by [Chen and Goodman, 1999] are as follows: firstly, interpolation performs better on small training sets , while Katz smoothing performs better on large training sets; secondly, absolute discounting is superior to linear discounting; thirdly, interpolated models are superior to back-off models for low (nonzero) counts; fourthly, adding free parameters to a smoothing algorithm and optimizing these parameters on held-out data can improve the performance, e.g., modified Kneser-Ney smoothing; and finally, interpolated Kneser-Ney and modified Kneser-Ney smoothing are the best and state-of-the-art smoothing algorithms.

Although the smoothed n-gram language model has been demonstrated to be a simple but effective model, the struggle to improve over it continues [Jelinek, 1991]. Broadly speaking, such attempts focus on the improved modelling of word sequences,

or on the incorporation of richer knowledge. In the following sections, we will review some extensions to an MLE-based n-gram model along these two directions: improvement over maximum likelihood estimation, and incorporation of richer knowledge.

2.2.2 Improvement over Maximum Likelihood Estimation

MLE-based LMs directly estimate language model probabilities using word frequencies. Smoothing is therefore required to obtain a better LM. There are various approaches which aim to improve on maximum likelihood estimation for n-gram LMs. These approaches, with different perspectives for language model smoothing, include distributed representation models [Bellegarda, 1998; Xu and Rudnicky, 2000; Bengio et al., 2001; Schwenk and Gauvain, 2002; Xu et al., 2003; Bengio et al., 2003; Morin and Bengio, 2005], latent variable models [Blitzer et al., 2005], and a Bayesian framework [MacKay, 2003; Teh, 2006b; Goldwater et al., 2006b]. We will introduce each of the three models in sequence.

2.2.2.1 Distributed Representation

Generalisation is another important problem relating to smoothing in n-gram language models. However, generalisation is intrinsically difficult for models that aim at learning the joint distribution among many discrete random variables like n-grams because of the *curse of dimensionality*: a word sequence on which the model will be tested is likely to be different from all the word sequences seen during training [Bengio et al., 2003]. Traditional but successful approaches for generalisations of n-grams are based on smoothing methods such as discounting, back-off and interpolation. Generalisation in discrete spaces, however, is not as obvious as in continuous spaces. Inspired by the relative easiness of generalisation in continuous spaces, approaches using distributed representations for language models have been proposed, in which distributed representations for words are learned. In this way, probabilities are more smoothly distributed in lower dimensional spaces, and each training sentence is allowed to inform the model about an exponential number of semantically neighbouring sentences [Bengio et al., 2003].

Language models based on neural networks [Xu and Rudnicky, 2000; Bengio et al., 2001; Schwenk and Gauvain, 2002; Xu et al., 2003; Bengio et al., 2003; Morin and Bengio, 2005], in which the neural network is used to learn the distributed representations for words, are among the family of distributed language models. One exam-

ple in this line of work is the neural probabilistic language model (NPLM) [Bengio et al., 2003]. The central idea of the approach is three-fold. Firstly, each word in the vocabulary is associated with a distributed word feature vector with m dimensions, which is a real-valued vector in $\mathbb{R}^m$. This can be regarded as a mapping C from any element w_i in vocabulary V to a real vector $C(w_i) \in \mathbb{R}^m$. Secondly, the joint probability function of word sequences $f(w_t, w_{t-1}, \ldots, w_{t-n+1}) = P(w_t | w_{t-n+1}^{t-1})$ is then expressed in terms of the feature vectors of those words in the sequence. Considering the constraints of probability theory (i.e., $\sum_{i=1}^{|V|} f(i, w_{t-1}, \ldots, w_{t-n+1}) = 1$ and $f > 0$), the joint probability function f can be decomposed into two parts, $f(i, w_{t-1}, \ldots, w_{t-n+1}) = g(i, C(w_{t-1}), \ldots, C(w_{t-n+1}))$, where the context is mapped to continuous feature vectors using *shared* mapping C, and g is a function that maps an input sequence of feature vectors for words in context to a conditional probability distribution over words in V for the next word w_t. The output of g is a vector whose i-th element estimates the probability $P(w_t = i | w_{t-n+1}^{t-1})$. Thirdly, the word feature vectors and the parameters of the probability function are learned simultaneously using neural networks, with a *softmax* output layer to guarantee positive probabilities summing to 1, $P(w_t | w_{t-1}, \ldots, w_{t-n+1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$.

Training for NPLM can be achieved by optimising a likelihood function using stochastic gradient ascent algorithms. Experiments on Brown corpus and Associated Press (AP) News have shown significant reductions in perplexity, ranging from 10% to 20% relative reduction in perplexity comparing to smoothed trigram using modified Kneser-Ney algorithm and class-based n-gram. Some reductions in word error rate (WER) using neural probabilistic language models have also been reported in [Schwenk and Gauvain, 2002, 2004; Emami and Jelinek, 2004; Emami and Mangu, 2007; Saon et al., 2010] on large-vocabulary continuous speech recognition. As claimed by [Bengio et al., 2003], the main reason for the perplexity reduction is that the neural probabilistic language model allows to take advantage of the learned distributed representation to fight the curse of dimensionality with its own weapons and consequently has good generalisations: by learning an embedding for words in a continuous space, each training sentence informs the model about a combinatorial number of other sentences. In this sense, the idea of distributed representations for words is similar to vector-space representation in information retrieval, such as latent semantic indexing (LSI) [Deerwester et al., 1990] that learns the distributed representations for documents and words by applying singular value decomposition (SVD) on the co-occurrence matrix of documents and words. This distributed representations have also been applied

to statistical language modelling [Bellegarda, 1998].

Despite parallel implementations [Bengio et al., 2003], and some speed-up optimizations using hierarchical decomposition [Morin and Bengio, 2005], distributed language models based on neural networks however still suffer from an extremely slow speed for both training and testing, in comparison to the traditional n-gram language models. The fact that NPLMs use softmax to compute the probability of a word w given its context makes necessary a separate normalization for each context: estimating the parameters of a softmax requires $O(|V|)$ computation per observed context where $|V|$ is the size of the vocabulary. This restricts the applications of NPLMs on large vocabulary ASR tasks. Schwenk and Gauvain approximated the NPLM by only considering a short-list of the top $N = 8192$ words to fulfill the constraints on the processing time of the NPLM [Schwenk and Gauvain, 2005].

It is argued that traditional n-gram models smoothed by simple back-off formulas or interpolation schemes do not typically represent or take advantage of statistical regularities among contexts [Blitzer et al., 2004]. One expects the probabilities of rare or unseen events in one context to be related to their probabilities in statistically similar contexts. It thus should be possible to estimate more accurate probabilities by exploiting these regularities.

Latent variable models, such as the aggregate Markov model (AMM) [Saul and Pereira, 1997] and more general aspect model [Hofmann and Puzicha, 1998], are approaches for sharing statistical information across contexts, by factoring the conditional probability of a word given its context by a latent variable representing context 'classes'.

$$P(w|\mathbf{u}) = \sum_z P(z|\mathbf{u})P(w|z) \tag{2.19}$$

where $\mathbf{u}$ is the context for predictive word w, and z is the common context class. By introducing the latent class variable z, the regularities among different contexts can therefore be shared for w. However, this latent variable approach is difficult to generalize to multi-word contexts, as the size of the conditional probability table for latent class z given context grows exponentially with the context length [Blitzer et al., 2004].

The distributed representation for words in NPLM, on the other hand, encodes words as continuous feature vectors. It is thus easy to represent the contexts as low-dimensional continuous feature vectors by simply concatenating in sequence the feature vectors for each word in contexts. Blitzer et al. [2004] utilize the idea of combining the advantages of the AMM and NPLM. This hierarchical distributed representation for

language model results in a model that exploits the regularities among contexts as in the AMM and extends to multi-word context as in the NPLM. Compared to the NPLM, this proposed model is more efficient, as it avoids the per-context normalization, one of the most intensive computations in NPLM. In this model, contexts are represented as low-dimensional real-value vectors initialized by applying unsupervised algorithms for nonlinear dimensionality reduction [Weinberger et al., 2004] on the bigram counts matrix whose element C_{ij} records the number of times that word w_j follows word w_i. The probabilities of words given contexts are represented by a hierarchical mixture of experts (HME) [Jordan and Jacobs, 1994], where each expert is a multinomial distribution over predictive words. A tree-structured mixture model, is used in HME, which allows a rich dependency on context without expensive per-context normalization. Simply considering the bigram case, the probability of a word w conditioned on a context word w' is computed as:

$$P(w|\mathbf{u}) = P(w|w') = \sum_{\pi} P(\pi|\vec{x}(w'))P(w|\pi) \tag{2.20}$$

where $\vec{x}(w')$ is the input vector obtained by nonlinear dimensionality reduction and thus a function of the context words (i.e., w' in this case). π denotes a path from root to leaf through the HME tree. This takes a similar form to Equation (2.19) for the AMM. The interior nodes of the tree perform binary logistic regressions on the input vector to the HME, while each leaf specifies a multinomial distribution over the predictive word w'. The mixture weight for each leaf is computed by multiplying the probabilities of each branch along the path to that leaf. The Expectation-Maximization (EM) algorithm can be used to learn the maximum likelihood parameters for the HME [Jordan and Jacobs, 1994]. Experiments on ARPA North American Business News (NAB) corpus shows the proposed model consistently matches or outperforms a baseline class-based bigram model.

2.2.2.2 Latent Variable Models

Another example in the line of work that combines distributed models and latent variable models is also proposed by [Blitzer et al., 2005]. Instead of using a path through the HME tree π to represent the latent variable z, the latent variable z is extended to a vector of m binary latent bit variables, $z \equiv \mathbf{b} = (b_1, \ldots, b_m)$. The variables b_i are conditionally independent given context $\mathbf{u}$, which leads to the advantage that its vector elements model independent properties of their inputs. Like the AMM and the aspect model, the probability of a word w given its context $\mathbf{u}$ can be expressed as

$P(w|\mathbf{u}) = \sum_{\mathbf{b}} P(\mathbf{b}|\mathbf{u})P(w|\mathbf{b})$. To further exploit the distributed nature of $\mathbf{b}$, a multinomial logistic (also known as conditional maximum entropy) model is selected for $P(w|\mathbf{b})$, that is, $P(w|\mathbf{b}) = \frac{1}{Z(\mathbf{b})} \exp \sum_{i=1}^{m} \psi(b_i, w)$. Together with the conditional independence assumption of b_i given $\mathbf{u}$, the target distribution can be specified as follows:

$$P(w|\mathbf{u}) = \sum_{\mathbf{b}} \prod_{i=1}^{m} P(b_i|\mathbf{u}) \frac{1}{Z(\mathbf{b})} \exp \sum_{i=1}^{m} \psi(b_i, w) \tag{2.21}$$

where $\psi(b_i, w)$ is assumed to be an arbitrary, real-valued function of b_i and w as in standard conditional maximum entropy models.

Rather than representing words as distributed forms, this so-called distributed Markov model (DMM) treats the latent variables as distributed representations. By introducing the intermediate distributed latent variable vectors, common regularities are supposed to be shared among different contexts. Actually, the role of distributed latent variable vectors $\mathbf{b}$ can be considered as a distributed *clustering* of the different contexts in the continuous space formed by the real-valued vectors $\mathbf{b} = (b_i, \ldots, b_m)$. In addition, this distributed model can be easily extended to a multiword context, by simply encoding each word in the context with a separate vector of latent variables. Parameters in the case of smaller value of m, the length of distributed latent vector $\mathbf{b}$, can be estimated by traditional EM and generalized iterative scaling (GIS) algorithms as in [Blitzer et al., 2005], as for a mixture distribution. However, the fact that the computation grows exponentially with respect to the number of latent variables in the DMM model makes the exact inference difficult for larger values of m, which consequently requires approximate inference techniques in this case. Experiments on the Penn Treebank Wall Street Journal corpus show that the DMM has the capability of inducing the latent classes among contexts. Perplexity results also indicate that the DMM outperforms the single latent variable AMM and trigram on the specific tasks and corpus.

Distributed representations for language modelling are interesting and attractive, in that they provide a low-dimensional continuous space where further operations such as clustering and generalisation can be flexibly carried out. Neural networks based approaches like NPLM are simple and straightforward ways to jointly learn the distributed representation for words together with the conditional probability table. However, considering the costly computations in NPLM, the distributed latent variable models are preferable. More generally, distributed latent variable models in [Blitzer et al., 2005] share the similar intuition that assumes a latent structure (class) underlying the contexts as topic models. The difference is that the latter is a generative model

in which the observed variables (bag-of-words) are generated from the latent variable, while the former (like the DMM) introduces latent variables to separate contexts and predictive words, and assumes the latent variables are distributed conditional on contexts. This conditional form makes it easier to incorporate the history information, comparing to topic models that are based on "bag-of-words" assumption and using only unigram models.

Supposing that multimodal cues could be mapped to a common continuous space as words, the conditional probabilities or co-occurrence statistics of interests can then be measured by simply using the Euclidean distances between different multimodal cues and words. This idea has been verified in the work on Euclidean embedding for co-occurrence data [Globerson et al., 2004]. This method embeds objects of different types, such as images and texts, into a single common Euclidean space based on their co-occurrence statistics. The joint distributions for objects are then modelled as exponentials of Euclidean distances in the low-dimensional embedding space. The authors have successfully demonstrated this method on NIPS database using word-document and author-document co-occurrence statistics, and claimed that this method outperforms standard methods of statistical correspondence modelling, such as multi-dimensional scaling and correspondence analysis.

2.2.2.3 Bayesian Interpretation

Bayesian models have begin attracting much attention recently, with the availability of more efficient approximate inference algorithms and more computation resources. In a Bayesian framework for language modelling, smoothing is taken into consideration naturally for language modelling via a parameterised prior distribution. By using a suitable prior distribution, Bayesian language models can provide better performance than state-of-the-art smoothed n-gram LMs for ASR. We will review current approaches to Bayesian language modelling in Section 2.3.3.

2.2.3 Explore Richer Knowledge

Traditional n-gram LMs rely on only the immediately proceeding $n-1$ lexical words. There are two issues limiting the performance of n-gram LMs. First, the history is short-span, i.e., only the $n-1$ previous words. Second, the history contains only lexical information. High level linguistic knowledge based on lexical words, for example, part-of-speech (POS), morphological information, syntax, and semantics, have

been widely studied before to enhance language modelling. On the other hand, non-linguistic knowledge, i.e., multimodal cues available from multiparty conversational meetings, are promising knowledge sources for language modelling. We briefly review the exploration of richer knowledge other than lexical information for language modelling in this section. Some knowledge can be used directly (e.g. POS tags), while others require statistical models to learn/extract from data (i.e., topics, semantic context).

2.2.3.1 Linguistic Knowledge

POS tags represent the syntactic roles of words. It is straightforward to incorporate POS tags with n-gram LMs. Jelinek [1990] initially used POS tags as classes in a conditional probabilistic model, i.e., first predicting POS tags from history and then predict next word based on the POS tag. Heeman and Allen [1997] used POS tags as part of the context for estimating language model probabilities using a decision tree, and found small reduction in perplexity. Heeman [1998] also compared this POS-based LM using decision trees with class-based LMs, and observed that the POS-based LM was slightly better than the class-based LM in perplexity. Using POS for language modelling has also been investigated in [Kirchhoff et al., 2003] within a factored language model. However, it is not so successful in utilizing POS information for language models.

A structured language model developed hidden hierarchical syntactic-like structure incrementally, and use it to extract meaningful information from the word history to complement the locality of n-gram LMs [Chelba and Jelinek, 1998, 2000]. It claimed to enable the use of long distance dependencies, by assigning probabilities to every joint sequence of words-binary-parse-structure with headword annotation and operating in a left-to-right manner. A structured language model will assign a probability $P(W,T)$ to every sentence W with every possible POS tag assignment, binary branching parse, non-terminal label and headword annotation for every constituent of T. Improvements over standard trigram LMs in perplexity and lattice rescoring were reported by using a structured language model. In [Xu et al., 2003], the components of a structured language model were modelled using connectionist models, and further trained by an EM procedure. Xu and Jelinek [2004] constructed Random Forests (RFs) by randomly growing decision trees using syntactic information, and explored the use of RFs in the structured language model. RFs have been shown to generalize well to unseen data. Compared to a baseline LM using Kneser-Ney smoothing, RFs using

syntactic information reduced perplexity, and word error rate in N-Best rescoring, on large vocabulary speech recognition for UPenn Treebank portion of the WSJ corpus.

An almost-parsing language model, called *Super*ARV, was proposed by [Wang and Harper, 2002] to incorporate syntactic constraints with lexical features, using Constraint Dependency Grammar (CDG) [Harper and Helzerman, 1995]. In *Super*ARV LM, words typically have a number of *Super*ARV tags to indicate different types of word usage. The authors reported reductions in perplexity and word error rate during lattice rescoring comparing to baseline trigram LMs and POS-based LMs, on Wall Street Journal Penn Treebank and CSR corpora.

Bilmes and Kirchhoff [2003] reported experimental results using syntactic information such as morphological class and stems in a factored language model (FLM), together with a generalized parallel backoff (GPB) algorithm. Results on the CallHome-Arabic corpus showed that FLSs with GPB could produce bigram LMs with comparable perplexity to baseline trigram LMs.

Syntax is the study of the symbols of a language, without referring to their meaning. Semantics, on the contrary, refers to high level knowledge – meaning – about a language. We will review the use of semantic knowledge for language modelling in Section 3.2.2.

In practice, incorporating linguistic knowledge into LMs is not trivial: First, those simple and ready-to-use forms of linguistic knowledge, such as POS tags, do not always work for an ASR system, although it is easier to obtain reductions in perplexity. Second, the use of syntactic and semantic knowledge normally requires complicated models, such as a parser, to extract the information from a word sequence. This constrains the applications of LMs augmented by these linguistic knowledge, for example, we normally have to use these LMs in the second pass decoding for ASR, i.e., N-Best or lattice rescoring.

2.2.3.2 Non-linguistic Knowledge

Considering one of the tasks of this thesis – to estimate an augmented language model for meeting ASR using non-linguistic knowledge, are there any novel multimodal cues that we can make use of to compensate the weakness of traditional n-gram LMs using only linguistic knowledge? Besides the text modality, group meetings have data coming from multiple speaker sources, and in multiple modalities, i.e., audio, video, and gesture. There have been some previous attempts to incorporate non-linguistic and multimodal knowledge into n-gram LMs, and we will review those attempts in Sec-

tion 3.2. One challenge for applying multimodal cues in n-gram LMs, including using prosody, social information, and visual cues, is that multimodal cues are normally heterogeneous, with different types and different scales from lexical words. Therefore we need to find statistical models to incorporate multimodal cues in LMs, such as the hierarchical Bayesian models proposed in this thesis.

2.3 Hierarchical Bayesian Model

Bayesian analysis explicitly uses probability to quantify degrees of belief. Bayes' theorem in Equation (2.22) explains the relationship between the *prior*, the *likelihood*, and the *posterior*, where θ denotes the unknown parameters from the sample space Θ, and $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ denotes the data from some sample space $\mathcal{X}$ (continuous or discrete):

$$P(\theta|\mathcal{D}) = \frac{P(\mathcal{D}|\theta)P(\theta)}{P(\mathcal{D})} = \frac{P(\mathcal{D}|\theta)P(\theta)}{\int_{\Theta} P(\mathcal{D}|\theta)P(\theta)d\theta} \,. \tag{2.22}$$

The prior $P(\theta)$ represents the belief in the parameters θ before observing the data $\mathcal{D}$, and the posterior $P(\theta|\mathcal{D})$ represents the updated belief in the parameters after having observed the data $\mathcal{D}$. The likelihood is often described using an exponential family distribution, which is mathematically convenient since conjugate priors exist for exponential family likelihood functions[1]. Bayesian modelling differs from MAP (maximum a posteriori) by computing the posterior distribution of θ rather than its maximum: this can be computationally challenging, due to the complexity of integrating over a large sample space Θ.

Currently, there is a resurgent interest in the Bayesian framework for machine learning and pattern recognition. This is partly attributed to the recent progress of probabilistic graphical models and development of efficient techniques for approximate inference. *Hierarchical Bayesian Models* [Gelman et al., 2004] have been well studied and enable the construction of richer statistical models in which the prior may depend on parameters that are not involved in the likelihood [Gelman et al., 2004]. The prior distribution $P(\theta)$ is itself a member of a family of densities with *hyperparameters* λ, $P(\theta|\lambda)$. The hierarchical Bayesian framework provides a better modelling of multi-parameter problems for hierarchical data, where non-hierarchical models usually tend to inappropriately overfit such data. Hierarchical Bayesian models can have enough parameters to fit the data well, while using a population distribution to structure some

[1] A prior distribution is conjugate to the likelihood functions if Bayes' theorem results in a posterior distribution from the same family as the prior.

dependence into the parameters, thereby avoiding problems of overfitting [Gelman et al., 2004].

The basic idea underlying hierarchical Bayesian modelling approaches is to estimate the joint posterior distribution of all the (normally unobserved) parameters in a latent variable model from a specific training dataset and compute the probability distributions of interest by marginalizing out the unobserved/uninterested parameters and finally test on generalized testing data set, all in a full Bayesian inference manner. Hierarchical Bayesian approaches are largely used to facilitate inferences in a *latent variable model*, which is in turn normally described by a graphical model where *nodes* denote random variables (with shaded nodes for observed variables and unshaded for unobserved/latent ones), *edges* possible dependencies between random variables, and *plates* replications of a substructure respectively.

2.3.1 Bayesian Priors

Bayesian analysis begins with a prior distribution capturing any available prior knowledge about the process generating the data. We will introduce several popular prior distributions in Bayesian data analysis, including Dirichlet distributions, Dirichlet processes, and Pitman-Yor processes.

2.3.1.1 Dirichlet Distribution

The Dirichlet distribution [Antoniak, 1974], a multivariate generalisation of the beta distribution, is a density over a $(K-1)$-simplex, i.e., K-dimensional vectors $\mathbf{p}$ whose components p_i are all non-negative and sum to 1, and is hence well suited to define a distribution over a multinomial distribution. The Dirichlet distribution is parameterised by a K-dimensional measure $\alpha\mathbf{m}$ where $\mathbf{m}$ is a normalized measure over K components ($\sum_i m_i = 1$) and α is a positive scalar. The Dirichlet distribution with parameters $\alpha\mathbf{m}$ has a probability density function given by:

$$P(\mathbf{p}|\alpha\mathbf{m}) = \frac{1}{Z(\alpha\mathbf{m})}\prod_{i=1}^{K} p_i^{\alpha m_i - 1} \tag{2.23}$$

with normalisation constant $Z(\alpha\mathbf{m}) = \prod_{i=1}^{K}\Gamma(\alpha m_i)/\Gamma(\alpha)$ where $\Gamma()$ is the gamma function. The vector $\mathbf{m}$ represents the mean of the Dirichlet distribution, and α is a concentration parameter with larger values of α drawing samples away from the corners of the simplex to the centre, peaking around the mean $\mathbf{m}$. One reason why the

Dirichlet distribution is selected as a prior is because it is *conjugate* to the multinomial distribution: i.e., the posterior is also a Dirichlet distribution when the prior is a Dirichlet distribution and the likelihood is a multinomial. We use $\mathrm{Dir}(\alpha\mathbf{m})$ to denote a Dirichlet density with hyperparameters $\alpha\mathbf{m}$. When $K = 2$, the Dirichlet distribution is equivalent to the beta distribution.

Figure 2.2, copied from [Sudderth, 2006], shows examples of four Dirichlet densities with different parameters on the simplex $(\pi_1, \pi_2, 1-\pi_1-\pi_2)$. We see in the figure the effect of concentration parameter α – with larger values of α drawing samples away from the corners of the simplex to the centre, peaking around the mean $\mathbf{m}$.

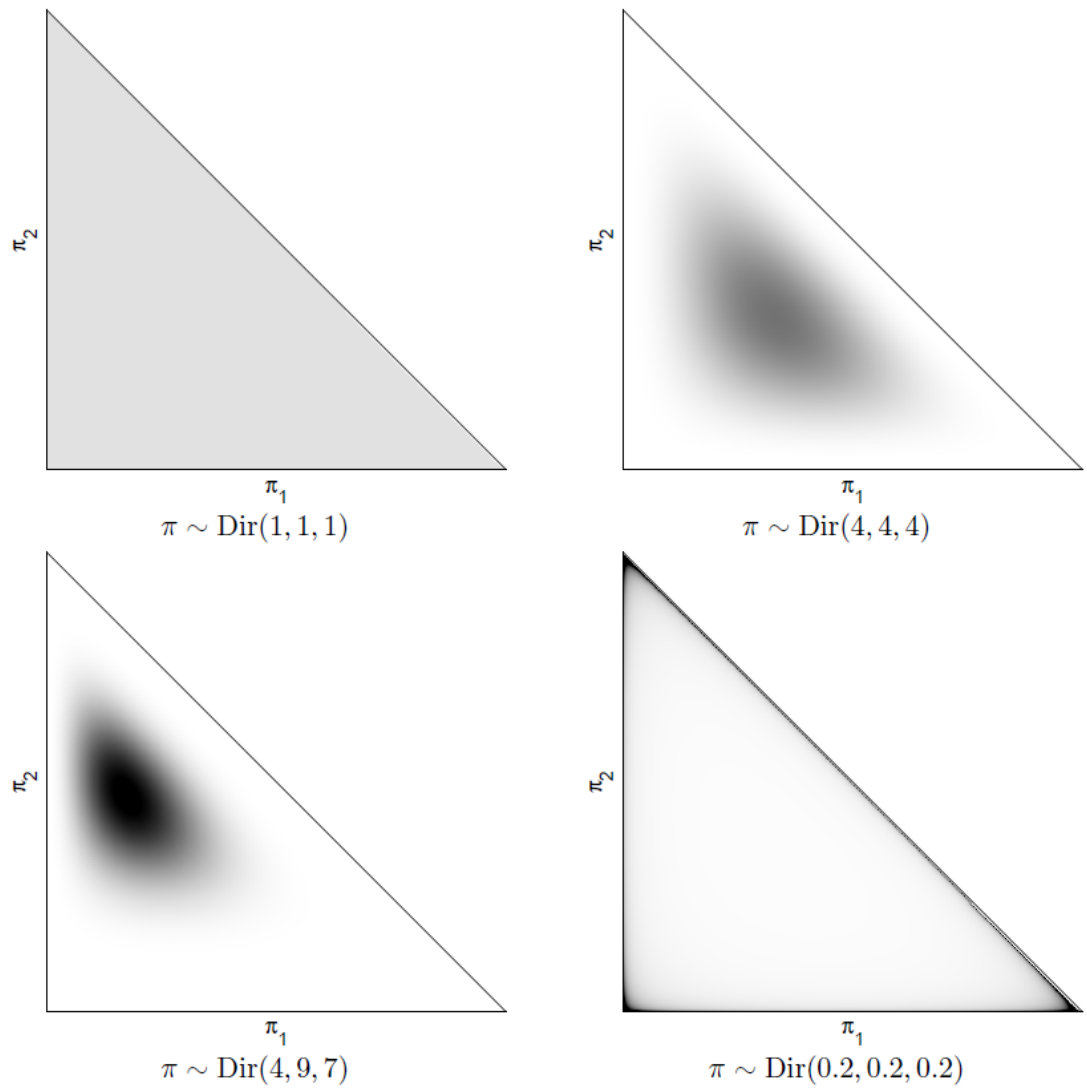


Figure 2.2: Dirichlet densities with $K = 3$ on the simplex $(\pi_1, \pi_2, 1-\pi_1-\pi_2)$. Darker intensities indicate regions with higher probability. This figure was copied from [Sudderth, 2006].

2.3.1.2 Dirichlet Process

The Dirichlet process (DP) is a stochastic process, first formalised in [Ferguson, 1973] for general Bayesian modelling, which has become an important prior distribution for nonparametric models. Nonparametric models are characterised by allowing the number of model parameters to grow with the amount of training data. This helps to alleviate over- or under-fitting problems, and provides an alternative approach to parametric model selection or averaging.

A random distribution G over a space Θ is called a Dirichlet process distributed with base distribution H and strength or concentration parameter α, if

$$(G(A_1),\ldots,G(A_r)) \sim \mathrm{Dir}(\alpha H(A_1),\ldots,\alpha H(A_r)) \tag{2.24}$$

for every finite measurable partition $A_1,\ldots,A_r$ of Θ [Ferguson, 1973]. We write this as $G \sim \mathrm{DP}(\alpha,H)$, and it may be interpreted as a distribution over distributions. The parameter H, a measure over Θ, is intuitively the mean of the DP. The parameter α, on the other hand, can be regarded as an inverse variance of its mass around the mean H, with larger values of α for smaller variances. More importantly in infinite mixture models, α controls the expected number of mixture components in a direct manner, with a larger α implying a larger number of mixture components a priori.

Draws from an DP are composed as a weighted sum of point masses located at the previous draws $\theta_1,\ldots,\theta_n$. This leads to a constructive definition of the DP called the stick-breaking construction [Sethuraman, 1994]. If we construct G as follows:

$$\begin{aligned} \beta_k &\sim \mathrm{Beta}(1,\alpha) & \theta_k^* &\sim H \\ \pi_k &= \beta_k \prod_{l=1}^{k-1}(1-\beta_\ell) & G &= \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k^*} \end{aligned} \tag{2.25}$$

Then $G \sim \mathrm{DP}(\alpha,H)$. θ_k^* is a unique value among $\theta_1,\ldots,\theta_n$, and $\delta_{\theta_k^*}$ denotes a point mass at θ_k^*. The construction of π can be understood as follows [Teh, 2007]. Starting with a stick of length 1, first break it at β_1, assign π_1 to be the length of stick just broken off. Then recursively break the other portion to obtain π_2,π_3 and so forth. The stick-breaking distribution over π satisfies $\sum_{k=1}^{\infty}\pi_k = 1$ with probability one. This definition is important for inference of an DP.

The procedure for generating draws from an DP can also be interpreted using the Chinese restaurant franchise [Teh, 2007].

Given observed values of $\theta_1,\ldots,\theta_n$, the posterior distribution of G is again distributed according to another DP with updated hyperparameters, with the following

form [Teh, 2007]:

$$G|\theta_1,\ldots,\theta_n \sim \mathrm{DP}\left(\alpha+n, \frac{\alpha}{\alpha+n}H + \frac{n}{\alpha+n}\frac{\sum_{i=1}^{n}\delta_{\theta_i}}{n}\right), \tag{2.26}$$

where the posterior base distribution is a weighted average between the prior base distribution H and the empirical distribution $\sum_{i=1}^{n}\delta_{\theta_i}/n$.

2.3.1.3 Pitman-Yor Process

The Pitman-Yor process or the two-parameter Poisson-Dirichlet process [Pitman and Yor, 1997; Pitman, 1995] $\mathrm{PY}(d,\theta,G_b)$ is a three parameter distribution over distributions, where d is a discount parameter, θ a strength parameter, and G_b a base distribution that can be understood as a mean of draws from $\mathrm{PY}(d,\theta,G_b)$. When $d=0$, the Pitman-Yor process reverts to the Dirichlet process $\mathrm{DP}(\theta,G_b)$. In this sense, the Pitman-Yor process is a generalisation of the Dirichlet process.

The procedure for generating draws $\mathrm{G} \sim \mathrm{PY}(d,\theta,G_b)$ from a Pitman-Yor process can be described using the "Chinese Restaurant" metaphor [Aldous, 1985; Pitman, 1995]. Imagine a Chinese restaurant containing an infinite number of tables, each with infinite seating capacity. Customers enter the restaurant and seat themselves. The first customer sits at the first available table, while each of the subsequent customers sits at an occupied table with probability proportional to the number of customers already sitting there $c_k - d$, or at a new unoccupied table with probability proportional to $\theta + dt_\bullet$, where $t_\bullet$ is the current number of occupied tables. That is, if z_i is the index of the table chosen by the ith customer, then the ith customer sits at table k given the seating arrangement of the previous $i-1$ customers $\mathbf{z}_{-i} = \{z_1,\ldots,z_{i-1}\}$ with probability

$$P(z_i = k|\mathbf{z}_{-i}) = \begin{cases} \frac{c_k - d}{\theta + c_\bullet} & 1 \le k \le t_\bullet \\ \frac{\theta + dt_\bullet}{\theta + c_\bullet} & k = t_\bullet + 1 \end{cases} \tag{2.27}$$

where c_k is the number of customers sitting at table k and $c_\bullet = \sum_k c_k$ is the total number of customers. The Pitman-Yor process with parameters (d,θ,G_b) produces a power-law distribution with index $1+d$ over the number of customers seated at each table [Goldwater et al., 2006b]. The power-law distribution — a few outcomes have very high probability and most outcomes occur with low probability — has been found to be one of the most striking statistical properties of word frequencies in natural language.

2.3.2 Bayesian Inference

Bayesian inference refers to the statistical process of updating the probability of outcomes/events using evidence or observations. Exact inference is normally computationally expensive, and often intractable, for many hierarchical Bayesian models. We instead resort to approximate inference algorithms.

In general, there are two types of approximate inference algorithms in Bayesian analysis for the estimation of posterior distributions of interest: Monte Carlo methods and variational methods. Monte Carlo methods [Andrieu et al., 2003; Gelman et al., 2004] use random samples to simulate probabilistic models. They are guaranteed to give arbitrarily precise estimates with sufficient computation. *Markov chain Monte Carlo (MCMC)* methods are a family of iterative Monte Carlo algorithms that draw samples from an otherwise intractable target density via a first-order Markov process. Gibbs sampling, a special case of the Metropolis-Hastings algorithm [Andrieu et al., 2003], is one of the most widely used MCMC methods for Bayesian inference. It assumes that it is tractable to sample from the conditional distribution of one of these variables given the other $(N-1)$ ones. We will use Gibbs sampling methods for inference in this thesis.

Variational methods [Frey and Jojic, 2005; Jordan et al., 1999], on the other hand, are a class of deterministic approximations to the problems of learning and inference for Bayesian inference. A variational method begins by expressing a statistical inference task as the solution to a mathematical optimization problem [Sudderth, 2006]. By approximating or relaxing the objective function, one can derive computationally tractable algorithms which bound or approximate the statistics of interest. Sudderth [2006] gives a good review and comparison of various Bayesian inference algorithms.

2.3.3 Bayesian Language Modelling

2.3.3.1 An Overview

In language models, we are interested in the conditional probabilities $P(w_i|\mathbf{u})$. Let $p_{\mathbf{u}}$ represent the vector parameterizing the conditional distribution for context $\mathbf{u}$, and $\Theta_{\mathbf{u}}$ represent the set of $p_{\mathbf{u}}$ for all contexts. The Bayesian estimation of LMs supposes that the training corpus is produced by a generative process, where words are independently and identically generated from a chosen distribution of $p_{\mathbf{u}}$, which in turn is drawn once for $\mathbf{u}$ in the whole data set. Under these assumptions, a general framework of Bayesian

estimation of LMs can be explained as follows.

- Assume $P(\Theta_{\mathbf{u}})$ is a prior distribution over $\Theta_{\mathbf{u}}$. Following the Bayesian formalism, the posterior of parameters $\Theta_{\mathbf{u}}$ is obtained as

$$P(\Theta_{\mathbf{u}}|w_1,\ldots,w_M) = \frac{P(w_1,\ldots,w_M|\Theta_{\mathbf{u}})P(\Theta_{\mathbf{u}})}{P(w_1,\ldots,w_M)} \tag{2.28}$$

- According to the independent and identically distributed (i.i.d.) assumption, the likelihood of parameters $\Theta_{\mathbf{u}}$ given the observed data $(w_1,\ldots,w_M)$ is

$$P(w_1,\ldots,w_M|\Theta_{\mathbf{u}}) = \prod_{i=1}^{M} P(w_i|\mathbf{u}_i,\Theta_{\mathbf{u}}) \tag{2.29}$$

- The evidence (or marginal likelihood) $P(w_1,\ldots,w_M)$ is obtained from the joint distribution $P(\Theta_{\mathbf{u}},w_1,\ldots,w_M) = P(\Theta_{\mathbf{u}})P(w_1,\ldots,w_M|\Theta_{\mathbf{u}})$ by integrating over the parameters $\Theta_{\mathbf{u}}$

$$P(w_1,\ldots,w_M) = \int \mathrm{d}\Theta_{\mathbf{u}} P(\Theta_{\mathbf{u}},w_1,\ldots,w_M) \tag{2.30}$$

- The predictive probability for the next word w_{M+1} the quantity in which we are most interested in LMs, $P(w_{M+1}|w_1,\ldots,w_M)$, can be derived by applying the sum rule of probability $P(C|A) = \int P(C|A,B)P(B|A)\mathrm{d}B$ to marginalize over unknown parameters $\Theta_{\mathbf{u}}$

$$P(w_{M+1}|w_1,\ldots,w_M) = \int \mathrm{d}\Theta_{\mathbf{u}} P(w_{M+1}|\Theta_{\mathbf{u}},w_1,\ldots,w_M)P(\Theta_{\mathbf{u}}|w_1,\ldots,w_M) \tag{2.31}$$

We can see from the above discussion that there are two key ideas for Bayesian approaches to language modelling. The first one is to place a prior over the probability distribution for LMs, and iteratively update the posterior distribution to represent the newly available knowledge from observed data so far, and gradually approach the true distribution for parameters of the LM as more and more data is seen. The second idea is to integrate, or marginalize, out those uninteresting parameters, to obtain the predictive probabilities of interest.

The idea of placing a prior distribution over parameters of LMs and learning point estimates of parameters from training data was investigated by Nadas in 1984 [Nadas, 1984]. However, this was an "empirical Bayes" perspective in which parameters of the prior were point estimates learned by maximizing the likelihood on the training data rather than by full Bayesian inference.

We need to apply some priors in the full Bayesian models. We will employ two priors: Dirichlet introduced in Section 2.3.1.1, and Pitman-Yor process introduced in Section 2.3.1.1 for language modelling.

2.3.3.2 Hierarchical Dirichlet Language Models

MacKay and Peto [1994] introduced a full Bayesian approach for language modelling, which is an extension of the simple Bayesian framework discussed above to the hierarchical Dirichlet language model (HDLM). The predictions of hierarchical Dirichlet LMs are similar to those of the traditional language modelling procedure known as 'smoothing'. They compared the new way of estimating LM probabilities using a prior distribution with conventional bigram models, which are specified by a conditional distribution $P(w_t = i|w_{t-1} = j)$ described by $V(V-1)$ (V is the number of vocabulary) free parameters and denoted by a $V \times V$ matrix $\mathbf{Q}$ with $P(w_t = i|w_{t-1} = j) \equiv q_{i|j}$. The jth row of matrix $\mathbf{Q}$ then represents the conditional probability vector for transitions from word j, denoted by a vector $\mathbf{q}_j$. To perform Bayesian inference, a Dirichlet prior, given an *unknown* measure over words $\beta\mathbf{m}$ (with $\beta > 0$ and $\sum_i m_i = 1$), is placed over the vectors $\mathbf{q}_j$ in $\mathbf{Q}$, representing our uncertainty about values for the parameters of conditional distributions in bigram models.

$$P(\mathbf{Q}|\beta\mathbf{m}) = \prod_j Dirichlet(\mathbf{q}_j|\beta\mathbf{m}) \tag{2.32}$$

In [MacKay and Peto, 1994], the predictive probability $P(i|j,D,\beta\mathbf{m})$ was derived by Bayes' theorem, combining the Dirichlet prior in Equation (2.32), likelihood $P(D|\mathbf{Q}) = \prod_j \prod_i q_{i|j}^{F_{i|j}}$ ($F_{i|j}$ is the conditional count of occurrence of $w_j w_i$ in data D), and the fact the posterior $P(\mathbf{Q}|D,\beta\mathbf{m})$ is again a Dirichlet distribution parameterised by $\mathbf{F} + \beta\mathbf{m}$.

$$P(i|j,D,\beta\mathbf{m}) = \frac{F_{i|j} + \beta m_i}{F_j + \beta} = \lambda_j m_i + (1-\lambda_j) f_{i|j} \tag{2.33}$$

where $f_{i|j} = F_{i|j}/F_j$ and $\lambda_j = \frac{\beta}{F_j+\beta}$. Notice the differences between Equation (2.33) and the conventional form of *deleted interpolation* [Jelinek and Mercer, 1980] smoothing method, i.e., $P(w_t = i|w_{t-1} = j) = \lambda f_i + (1-\lambda) f_{i|j}$ where $f_i = F_i/T$ and $f_{i|j} = F_{i|j}/F_j$ are the maximum likelihood estimators. Two points concerning the differences should be noted here. Firstly, instead of using marginal statistics f_i in deleted interpolation, m_i, the ith element of the Dirichlet prior parameters $\beta\mathbf{m}$, is now taking the same role. Secondly, the coefficient λ in Equation (2.33) varies inversely with the frequency of the given context j, which approximates the optimal coefficients in deleted interpolation

obtained by dividing contexts j into different groups according to frequency F_j and sharing the coefficient, optimized by cross-validation, within each group. MacKay and Peto [1994] in this way demonstrated the comparable performance of a hierarchical Dirichlet language model with that of a bigram model smoothed by deleted interpolation with specific values of λ on a small corpus. The smoothing that results from the HDLM is equivalent to the add-δ smoothing [Laplace, 1825; Lidstone, 1920; Johnson, 1932; Jeffreys, 1948] introduced in Section 2.2.1.1, with the psudocount δ comes from the prior Dirichlet distribution. This differs from absolute discounting approaches which subtract discounts from MLE counts. The HDLM, therefore, can not produce comparable results to smoothing approaches such as absolute discounting.

Recalling the assumption we made on $\beta\mathbf{m}$, the $\beta\mathbf{m}$ in Equation (2.33), however, is still assumed to be *unknown*. MacKay and Peto [1994] further assumed an uninformative prior $P(\beta\mathbf{m})$ over the measure, and approximated the posterior of $P(\beta\mathbf{m}|D)$ with maximum $[\beta\mathbf{m}]^{MP}$ using "empirical Bayes" procedure. This resulted in a *hierarchical Bayesian model* [Gelman et al., 2004], which is distinguished from "empirical Bayes" approaches in that Bayesian inference is used to control the parameters $\beta\mathbf{m}$ (called *hyperparameters* in hierarchical Bayesian models). Alternative options are available for the priors over $\beta\mathbf{m}$, another Dirichlet distribution with parameters $\alpha\mathbf{u}$, $P(\mathbf{m}|\alpha\mathbf{u}) = \frac{1}{Z(\alpha\mathbf{u})}\prod m^{\alpha u_i - 1} \equiv Dirichlet(\mathbf{m}|\alpha\mathbf{u})$, would give more informative knowledge for $\beta\mathbf{m}$. Moreover, this informative Dirichlet prior over $\beta\mathbf{m}$ can be considered as a way to share knowledge among contexts, in the spirit of smoothing methods for LMs.

One advantage of the hierarchical Dirichlet language model is that the assignment of zero probability mass is always avoided due to the constraint of the Dirichlet prior $Dirichlet(\mathbf{q}_j|\beta\mathbf{m})$ over words. In addition, in cases where only limited training data are available, prior knowledge can be applied to those insufficient observations, i.e., giving a uniform distribution for sparse contexts by providing a symmetric base measure $\mathbf{m}$. In cases where relatively sufficient training data are available, the resulting distribution on the other hand tends towards the maximum likelihood estimation for LMs. In this sense, the conditional distributions of LMs estimated by the Bayesian approach are smoother by nature than LMs obtained using MLE.

Cowans [2006] pointed out the similar role of the use of a shared Dirichlet prior ($Dirichlet(\mathbf{m}|\alpha\mathbf{u})$ in the above case) between hierarchical Dirichlet language models and topic models like latent Dirichlet allocation [Blei et al., 2003]. The shared prior in hierarchical Dirichlet LMs allows information to be shared concerning the distribution

over words in different contexts, whereas in topic models the shared prior allows information to be shared between documents concerning the distribution over topics. The similarity and relationship between hierarchical Dirichlet LM and topic model have been utilized by Wallach to explore her beyond "bag-of-words" topic modelling [Wallach, 2006].

2.3.3.3 Hierarchical Pitman-Yor Process Language Models

The selection of prior distributions becomes one of the central issues in Bayesian models. Priors vary according to different specific models and applications. The use of a Dirichlet prior in Bayesian models, for instance, has no strong justification other than its conjugacy to a multinomial distribution, which in turn makes it convenience for mathematical computation [Gelman et al., 2004]. Goodman also argued that a Dirichlet prior, as the assumption for proving the optimality of smoothing technique for language modelling in [MacKay and Peto, 1994], "is a bit odd", and does not seem to correspond well to reality according to their empirical work on language modelling [Goodman, 2001]. It is therefore natural to consider other potential prior distributions for Bayesian models. An alternative prior is the two-parameter Poisson-Dirichlet process or the *Pitman-Yor process* [Pitman and Yor, 1997; Ishwaran and James, 2001; Pitman, 2002], a nonparametric generalisation of the Dirichlet process.

Goldwater et al. [2006b] argued that a Pitman-Yor process is more suitable as a prior distribution than a Dirichlet distribution to applications in natural language processing, as the power-law distributions of word frequencies [2] produced by Pitman-Yor processes more closely resemble those phenomenon seen in natural languages. Motivated by the striking property of power-law distributions in natural languages, Goldwater et al. [2006b] proposed a general two-stage framework for language models, where a *generator* based on an underlying generative model for producing a set of words, and an *adaptor* using a Pitman-Yor process as the prior to transform the stream of words produced by the generator into one whose frequencies obey a power-law distribution. The authors empirically justified the role of word types in formal analyses of natural languages by using a morphology model as the generator and an adaptor based on a Pitman-Yor process in their experiments on unsupervised learning of the morphological structure of English, even though the strength and performance of this two-stage

[2] Based on the principles of *preferential attachment* and "rich-get-richer", the probability that a word w will occur with frequency n_w in a sufficiently large corpus is proportional to n_w^{-g} where g is a power-law parameter

framework for more general tasks have not been verified. It is also worth mentioning that another contribution of [Goldwater et al., 2006b] is the justification of the direct correspondence between their model and interpolated Kneser-Ney smoothing [Kneser and Ney, 1995] for conventional n-gram.

Another more recent work along the line of using Pitman-Yor processes in hierarchical Bayesian models for language modelling has been independently proposed by [Teh, 2006b]. It can be considered as a natural generalisation of the hierarchical Dirichlet language model by [MacKay and Peto, 1994] by using a Pitman-Yor process rather than the Dirichlet distribution, and both hierarchical and experimental extensions to a Pitman-Yor language model by [Goldwater et al., 2006b]. The experiments on APNews[3] shows that the novel hierarchical Pitman-Yor language model produces results superior to hierarchical Dirichlet language models and n-gram smoothed by interpolated Kneser-Ney, and comparable to those smoothed by modified Kneser-Ney [Chen and Goodman, 1999]. The results are more encouraging than other Bayesian approaches to language modelling, which in turn further motivates the adoption of hierarchical Bayesian models for language modelling.

As in [Goldwater et al., 2006b], Teh again shows that interpolated Kneser-Ney can be interpreted as a particular approximate inference scheme in the hierarchical Pitman-Yor language model. This interpretation is more useful than past ones, which are typically based on *ad hoc* and empirical justifications, as it can recover the exact formulation of interpolated Kneser-Ney and actually produces superior results.

We are interested in this Bayesian interpretation for language models, because that hierarchical Bayesian language models can be used as a smoothing approach but also can share information among contexts for LMs, and can be easily integrated with other probabilistic models such as hierarchical Bayesian topic models [Wallach, 2006]. More generally, Bayesian probabilistic models are internally coherent, have explicitly declared prior assumptions, and can be improved by incorporating additional knowledge sources or being included in larger probabilistic models in a principled manner [Teh, 2006b].

We will discuss further to the hierarchical Pitman-Yor process LM, its application to language modelling for ASR, and its extensions, in Chapter 4.

[3] Associated Press (AP) News from 1995 to 1996, with sizes for training, validation, test, and vocabulary sets as 14 million, 1 million, 1 million, and 17964 respectively.

2.3.4 Probabilistic Topic Modelling

2.3.4.1 An Overview

Nowadays it becomes more and more demanding for automatic approaches to analyse large quantities of unstructured data, such as documents, images, and audio archives. One popular idea for modelling these large collections of discrete data is based on latent variable and generative models, assuming there are some latent structures (i.e., of meaning, semantics) underlying the data, and from which data is subsequently generated. Latent structures, expressed by unobserved variables, can eventually be 'reconstructed' from data via probabilistic inferences. This process corresponds to finding short descriptions of the data, while preserving the essential statistical regulations. Significant progress, notably based on approaches to dimensionality reduction and matrix factorisation (i.e., *tf-idf* scheme, latent semantic indexing) in the field of information retrieval [Salton and McGill, 1983; Hofmann, 1999], has been made on this problem. Topic models, which have received a growing interest in the machine learning community, aim to find a latent representation connecting documents and words — the topic. In a topic model, words in a document exchangeably co-occur with each other according to their semantics, following the "bag-of-words" assumption.

Suppose there are D documents in the corpus, and W words in the vocabulary. Each document $d = 1, \ldots, D$ in the corpus is represented as a mixture of latent topics, with the mixing proportions over topics denoted by θ_d. Each topic $k = 1, \ldots, K$ in turn is a multinomial distribution over words in the vocabulary, with the vector of probabilities for words in topic k denoted by ϕ_k.

In this section, we focus on the review of two "bag-of-word" models, LDA and the HDP, following [Teh et al., 2006; Teh, 2007; Teh et al., 2008].

The parameters $(\theta_1, \ldots, \theta_J)$ are exchangeable in their joint distribution if $P(\theta_1, \ldots, \theta_J)$ is invariant to permutations of the indexes $(1, \ldots, J)$. In a hierarchical Bayesian model, a prior/population distribution (i.e., $P(\theta|\phi)$) is normally placed on θ and each parameter θ_j is assumed as an independent sample from the prior distribution $P(\theta|\phi)$ governed by some unknown (hyper)parameter vector ϕ. The unknown hyperparameters ϕ in turn has its own prior distribution, called *hyperprior*, $P(\phi)$ governed by other parameters. In this sense, exchangeability indeed represents the *conditional* i.i.d. assumption in hierarchical Bayesian models. The joint probability distribution for θ is thus expressed as $P(\theta|\phi) = \prod_{j=1}^{J} P(\theta_j|\phi)$. By averaging over our uncertainty in unknown ϕ, the joint

probability distribution for θ in a hierarchical Bayesian model is given by

$$P(\theta) = \int \left[\prod_{j=1}^{J} P(\theta_j|\phi) \right] P(\phi) d\phi \tag{2.34}$$

In the infinite case where $J \to \infty$, *de Finetti's theorem* states that a collection of infinitely exchangeable random variables has the i.i.d. mixture representation, conditioned on another random parameter, for the joint probability distribution over the random variables in question, like in Equation (2.34). The above framework, described according to the notations in [Gelman et al., 2004], is the basic 'hierarchical' part in a hierarchical Bayesian model, which can be further repeated recursively to make more complicated full Bayesian models.

2.3.4.2 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) model [Blei et al., 2003], a hierarchical Bayesian topic model, is a mixture model subject to the exchangeability assumption and de Finetti's theorem. LDA explicitly assumes the exchangeability of words in a document, which results in a "bag-of-words" model where the order of words are ignored. In addition, documents in a corpus are implicitly assumed exchangeable in LDA. According to the exchangeability assumption for words in a document and the de Finetti theorem, LDA naturally takes a mixture form for the joint probability distribution over words in a document. More precisely, LDA can be regarded as a two-level hierarchical Bayesian model, where each document in a corpus is represented as a mixture of latent topics, and each topic in turn is represented as a mixture of words. Following the notations in [Blei et al., 2003], a generative procedure is used to describe the hierarchical Bayesian model for each document in a corpus:

1. **A prior distribution:** choose mixture weights θ, which are in a convex combination (i.e., a weighted sum whose weighting proportion coefficients θ_i sum to one), for topic mixtures according to the distribution $P(\theta)$.

2. **A generative process:** for each of the N words $\mathbf{w}$ in the document, choose a word w_n from the selected mixture configuration θ of topics according to the distribution $P(w_n|\theta)$.

Due to the exchangeability assumption for words in a document, the joint probability distribution over words in a document, by marginalizing out the mixture weights θ, is

then expressed as

$$P(w) = \int \left[\prod_{i=1}^{N} P(w_i|\theta) \right] P(\theta) \mathrm{d}\theta \tag{2.35}$$

which has a mixture representation like Equation (2.34) and thus follows the de Finetti theorem. Further, a Dirichlet prior with hyperparameter α is placed over the unknown parameters θ, i.e., $P(\theta|\alpha) = Dirichlet(\theta|\alpha)$. For conveniences, topic assignments $\mathbf{z}$ distributed according to a multinomial distribution over θ for N words are introduced, each of which is an index to the matrix β with row vector β_{z_i} as the multinomial distribution over words for topic z_i. $P(w_i|\theta)$ is therefore reformulated as $P(w_i|\theta) = \sum_{z_i} P(w_i|z_i, \beta) P(z_i|\theta) \equiv \sum_{z_i} P(w_i|z_i) P(z_i|\theta)$. The marginal distribution of a document is then obtained as

$$P(w|\alpha, \beta) = \int \left[\prod_{i=1}^{N} \sum_{z_i} P(w_i|z_i, \beta) P(z_i|\theta) \right] P(\theta|\alpha) \mathrm{d}\theta \tag{2.36}$$

Latent Dirichlet allocation [Blei et al., 2003] is a three-level hierarchical Bayesian model, which pioneered the use of the Dirichlet distribution for latent topics. That is, the topic mixture weights θ_d for the dth document are drawn from a prior Dirichlet distribution with parameters α, π:

$$P(\theta_d|\alpha\pi) = \frac{\Gamma(\sum_{i=1}^{K} \alpha\pi_i)}{\prod_{i=1}^{K} \Gamma(\alpha\pi_i)} \theta_1^{\alpha\pi_1 - 1} \ldots \theta_K^{\alpha\pi_K - 1} \tag{2.37}$$

where K is the predefined number of topics in LDA, Γ is the Gamma function, $\alpha\pi = \{\alpha\pi_1, \ldots, \alpha\pi_K\}$ represents the prior observation counts of the K latent topics with $\alpha\pi_i > 0$: π is the corpus-wide distribution over topics, and α is called the concentration parameter which controls the amount of variability from θ_d to their prior mean π.

Similarly, Dirichlet priors are placed over the parameters ϕ_k with the parameters $\beta\tau$. We write:

$$\theta_d|\pi \sim \mathrm{Dir}(\alpha\pi) \qquad \phi_k|\tau \sim \mathrm{Dir}(\beta\tau) \tag{2.38}$$

Figure 6.1.(A) depicts the graphical model representation for LDA. The generative process for words in each document is as follows: first draw a topic k with probability θ_{dk}, then draw a word w with probability ϕ_{kw}. Let w_{id} be the ith word token in document d, and z_{id} the corresponding drawn topic, then we have the following multinomial distributions:

$$z_{id}|\theta_d \sim \mathrm{Mult}(\theta_d) \qquad w_{id}|z_{id}, \phi_{z_{id}} \sim \mathrm{Mult}(\phi_{z_{id}}) \tag{2.39}$$

Based on the above description, the main objectives of inference in LDA are to find (1) the word distribution $P(\mathbf{w}|z_i, \beta)$ for each topic z_i, and (2) the topic distribution $P(\theta|\alpha)$ for each document. Since exact inference for the posterior distributions in LDA is intractable, a wide variety of approximate inference algorithms can be used for LDA, including Laplace approximation, mean-field variational methods [Jordan et al., 1999] such as simple convexity-based approach [Blei et al., 2003] and higher-order variational technique known as expectation propagation [Minka and Lafferty, 2002], and Markov-chain Monte Carlo (MCMC) methods such as Gibbs sampling [Pritchard et al., 2000] and collapsed Gibbs sampling [Griffiths and Steyvers, 2004]. The basic intuition underlying variational approximation methods is that complex graphs in graphical models can be probabilistically simple [Jordan et al., 1999]. The posterior distribution can thus be approximated by a simpler variational distribution optimized via a deterministic precess with a clear convergence criterion given by the tightest lower bound on the log likelihood. Advantages of MCMC algorithms like Gibbs sampling include their simplicity of implementation, and theoretical guarantees of convergence based on samples from the exact posterior. The disadvantages of MCMC approaches, however, are that these algorithms can be slow to converge to the stationary states of the Markov chain (burn-in period), and there is no well-defined criterion to assess the convergence.

It has been noted in literature that LDA closely relates to other probabilistic topic models, such as mixture of unigram [Nigam et al., 2000], Dirichlet mixture [Yamamoto and Sadamitsu, 2005], and pLSA [Hofmann, 1999]. For example, LDA is a continuous mixture of unigram model, and is a full Bayesian extension to pLSA by defining a complete generative model [Blei et al., 2003]. Girolami *et al.* showed that LDA with a uniform Dirichlet prior $Dirichlet(1)$ is a full Bayesian estimator for the same model for which pLSA provides an maximum likelihood or maximum a posterior (MAP) estimator [Girolami and Kabán, 2003]. Prior to the introduction of LDA to the text modelling field by Blei *et al.*, the work by Pritchard *et al.*, which is less known in text modelling field, has preempted LDA in its interpretation of a *admixture model*, which refers to a mixture whose components are itself mixtures of different features, for applications in genetics with different notations and terminologies [Pritchard et al., 2000]. More generally, Buntine and Jakulin presented a unified theory for models of analysing discrete data called *discrete component analysis* (DCA) [Buntine and Jakulin, 2006], which refers to models including pLSA, non-negative matrix factorisation (NMF), LDA, multinomial principal component analysis (MPCA) [Buntine, 2002], Gamma-Poisson

(GaP) model [Canny, 2004], and so on. The term *component* is used instead of *topic* as the notation for latent structure in discrete data. Buntine and Jakulin showed some interesting relationships within the DCA framework. For example, they showed the Dirichlet-multinomial version of DCA is actually an MPCA. By replacing the multinomial distribution of MPCA with a sequence of discrete distributions, we will reach an LDA. They also noted that both MPCA and LDA are special variations of an GaP, and that the Poisson version of LDA is an NMF model.

Basic topic models such as pLSA and LDA are based on the exchangeability assumption. The "bag-of-words" representation in topic models makes sense in some applications such as information retrieval and text classification, whereas it is not appropriate to neglect the word order information in other applications, i.e., language modelling for speech recognition. Therefore, constraining the exchangeability assumption (i.e., partial exchangeability) and moving beyond "bag-of-words" topic models is necessary if we want to take advantages of using both short-range information from local words and long-range information from global topics in language modelling. Griffiths *et al.* presented a composite generative model called HMM-LDA [Griffiths et al., 2004] that uses both short-range syntactic dependencies between words modelled by a HMM, and long-range semantic dependencies between words modelled by a LDA. HML-LDA, which is sensitive to word orders, is capable of simultaneously finding syntactic classes and semantic topics. Wallach proposed a hierarchical generative model that incorporates both n-gram statistics of a hierarchical Dirichlet bigram language model [MacKay and Peto, 1994] and latent topics of a LDA [Wallach, 2006]. This integration is totally within the hierarchical Bayesian framework. Both these extensions to LDA imply that it is possible for the marriage of language models, and topic models that go beyond the "bag-of-words" assumption, either by placing a Markov chain constrain over word sequences, or by full Bayesian ways.

While LDA was initially developed for document modelling, it has been widely used for other kinds of discrete data. The Author-topic (AT) model [Rosen-Zvi et al., 2004] is a generative model for documents that extends LDA to include authorship information. In AT, each author is represented by a multinomial distribution over topics, and each topic is represented as a multinomial distribution over words for that topic. A document with multiple authors is modelled as a distribution over topics that is a mixture of the distributions associated with the authors. AT therefore explores the relations between authors, documents, topic, and words. LDA has also been actively applied to image processing. Barnard *et al.* extended applications of LDA to multimodal data,

and proposed a generative model for images called the mixture of multi-modal LDA (MoM-LDA) [Barnard et al., 2003] to associate words with images. More complicated applications of LDA are learning and recognizing natural scene categories from static images [Li and Perona, 2005]. A category variable for classification was explicitly introduced to LDA, resulting in a hierarchical Bayesian model where the *topic* notation is replaced as *theme*. Another example is the usage of LDA for discovering objects and their location in images [Sivic et al., 2005]. Last but not least, LDA and its extensions have also been used for collaborative filtering [Blei et al., 2003], population genetics data [Pritchard et al., 2000], survey data [Erosheva et al., 2007], and social networks data [Airoldi et al., 2007].

A meeting is a dynamic process evolving over time. It is therefore desirable for topic models to exhibit dynamic characteristics. The use of a Dirichlet distribution in LDA to account for the topic mixture proportions makes it unable to model topic correlations. Blei *et al.* developed a correlated topic models (CTM) [Blei and Lafferty, 2005], where topic proportions exhibit correlation via a logistic normal distribution instead of the original Dirichlet distribution. The inference in CTM, however, is more complicated than LDA due to non-conjugacy of the logistic normal distribution to the multinomial distribution. A Dynamic topic model (DTM) [Blei and Lafferty, 2006] is an extension to LDA to model the time evolution of topics in large document collections. The approach to DTM is to use state space models that evolve with Gaussian noise on the natural parameters of the multinomial distributions that represent the topics and the mean parameters of the logistic normal distributions that represent the topic proportions. This a bit complicated topic model necessitates variational approximations based on Kalman filters and nonparametric wavelet regression for inference in DTM.

2.3.4.3 Hierarchical Dirichlet Process

Hierarchical Dirichlet process (HDP) [Teh et al., 2006] is a Bayesian nonparametric model that is useful to model multiple groups of data in which each group of data is associated with an underlying parameter and these parameters are shared together across groups. For example, the HDP allows us to share clusters across multiple clustering problems, or to share topics across multiple groups of documents.

We will describe the details about the HDP model in Chapter 6. We here show some applications of the HDP from previous research. Teh and Jordan [2010] gives a comprehensive introduction and review on the HDP and other hierarchical Bayesian

nonparametric models.

The HDP, a Bayesian nonparametric version of LDA, is useful for topic modelling [Teh et al., 2006]. The HDP extends finite mixture modelling in LDA to nonparametric setting in the HDP where the number of topics is automatically determined by the data. We use the HDP in this thesis for modelling topic and role information in multiparty meetings in Chapter 6.

Teh et al. [2006] also used the HDP to obtain an "infinite HMM" – an HMM with a countably infinite state space [Beal et al., 2002], and proposed the hierarchical Dirichlet process hidden Markov model (HDP-HMM). In the HDP-HMM, a state sequence is not drawn from a Markov chain over finite state space, but from an HDP to allow a countably infinite state space. Fox et al. [2008] proposed a special treatment to self-transitions by allowing a prior control of state persistence, and successfully applied the HDP-HMM to speaker diarization of meetings in which the number of participants is unknown a priori. They showed that the HDP-HMM approach yielded a state-of-the-art diarization method. The HDP-HMM has also been demonstrated to be useful for the problem of word segmentation [Goldwater et al., 2006a], and grammar induction [Johnson et al., 2007; Liang et al., 2007].

Cowans [2004, 2006] found information retrieval using the HDP improves upon state-of-the-art IR systems in relevance scores, and also showed that the HDP provides statistical justification for the intuition behind TF-IDF.

HDP has been used in statistical genetics for multi-population haplotype phrasing [Xing et al., 2006, 2007].

2.4 Summary

In this chapter, we reviewed some relevant statistical models. We begun with a brief introduction on automatic speech recognition, and then focus on a component of ASR – language model. We described the weakness of a MLE-based n-gram LM, and showed three different perspectives on language modelling. We reviewed in detail distributed representations and latent variable models for LMs. For distributed representations, it is a natural idea to incorporate multimodal cues in a distributed language model – if we could map various multimodal cues, including the lexical words, into to a common space, and carry out the inference and estimation on this common space. For latent variable models, they share the similar idea with topic models such as latent Dirichlet allocation – both try to learn some latent representations in between. For Bayesian

interpretation, we introduced the hierarchical Dirichlet language model, and leave the hierarchical Pitman-Yor process language model in Chapter 4. We also show some previous research on exploring richer knowledge in language models.

We introduced various prior distributions and Bayesian inference for hierarchical Bayesian models. We see how different priors could be used to obtain different Bayesian models for different tasks, for example, using Dirichlet distribution as prior in the HDLM and Pitman-Yor process in the HPYLM, for language modelling; and using Dirichlet distribution as prior in LDA and Dirichlet process in the HDP, for topic modelling.

Chapter 3

Multiparty Conversational Meetings

Since 2002, the U.S. National Institute of Standard and Technology (NIST) has been organizing the Rich Transcription (RT) evaluations, and advancing the development of the state-of-the-art automatic speech recognition technologies. The goal of the evaluation series is "to create recognition technologies that will produce transcriptions which are more readable by humans and more useful for machines"[1]. Two categories of tasks are broadly defined in the RT evaluations, Speech-to-Text Transcription (STT) tasks and Metadata Extraction (MDE) tasks. The STT tasks are the main evaluations in the RT evaluation series. For MED tasks, there are different sub-tasks, such as "Who Spoke When" speaker diarization, "Who Said What" speaker diarization, Speech Activity Detection, and Source Localization. A new task called Speaker Attributed STT (SASTT) has been introduced by merging STT and "Who Said What" speaker diarization since 2007. The task of the SASTT is to transcribe the spoken words and associate them with a speaker.

The NIST RT evaluation series first implemented STT tasks mainly for broadcast news speech and conversational telephone speech. In 2002, there was also a dry-run evaluation for meeting room speech in English. Since 2005, the RT evaluation series have focused on only English Meeting Domain speech.

There has been a growing research interest in the automatic transcription of multiparty conversational meetings since then, partially driven by the NIST RT evaluations. Examples include European-funded research projects such as AMI (Augmented Multi-party Interaction) project and its follow-up AMIDA (Augmented Multi-party Interaction with Distance Access)[2] [Renals et al., 2007], M4 (MultiModal Meeting Man-

[1] http://www.itl.nist.gov/iad/mig//tests/rt/
[2] http://www.amiproject.org

ager)[3], IM2 (Interactive Multimodal Information Management)[4], CHIL(Computers in the Human Interaction Loop)[5] [Waibel and Stiefelhagen, 2009]; and USA-funded research projects such as ICSI Meeting Recorder project [Morgan, 2001][6], CMU Meeting Record Creation and Access project [Waibel et al., 2001], CALO(Cognitive Assistant that Learns and Organizes)[7] [Tur et al., 2010].

The goal of this thesis is to investigate new approaches to augmenting language models for automatic transcription of multiparty conversational meetings, using better modelling techniques and richer knowledge available in meetings. We discussed the use of non-linguistic knowledge for language modelling in Section 2.2.3.2. In this chapter, we try to answer the question of what multimodal cues in multiparty conversational meetings are potentially useful for language modelling, based on the evidence from previous research in the literature. We will also introduce the meeting corpora and baseline ASR system used in this thesis.

3.1 Meeting Corpora

We will first introduce two multiparty conversational meeting corpora: the AMI Meeting Corpus, the main corpus used in this thesis, and the ICSI Meeting Corpus, which we use only in experiments of prosody for LMs in Chapter 7.

3.1.1 AMI Meeting Corpus

The main meeting corpus we use for the experiments described in this thesis is the AMI Meeting Corpus [Carletta, 2007], a freely available meeting corpus collected by the AMI project. There are in total about 100 hours of recorded meetings that are fully annotated. The meetings were recorded at three locations: Edinburgh, TNO, and Idiap. The participants, many of those are students, consist of both native and non-native English speakers.

The AMI Meeting Corpus is divided into two categories: scenario and non-scenario meetings. In the scenario meetings, four participants take part in each meeting and play roles within a fictional company. The scenario is that the four participants play different roles, project manager (the meeting leader), industrial designer, user-interface

[3] http://www.m4project.org
[4] http://www.im2.ch/
[5] http://chil.server.de
[6] http://www.icsi.berkeley.edu/Speech/mr/
[7] http://caloproject.sri.com

designer, and marketing expert, to design a new remote control in a company called Real Reactions. There are four meetings in each series of scenario meeting recording. The first meeting of each series is the kick-off meeting, where participants introduce themselves and become familiar with the task. The second is the functional design meeting, in which the team discusses the user requirements and determines the functionality and working design of the remote. The third meeting is the conceptual design of the remote, wherein the team determines the conceptual specification, the user interface, and the materials to be used. In the fourth and final meeting, the team determines the detailed design and evaluate their result.

For 5-fold cross-validation experiments in this thesis, we use only the scenario meetings from the AMI Meeting Corpus. In total there are 138 scenario meetings, with the length of each ranges from around 15 to 45 minutes.

The AMI corpus is freely available[8] and contains numerous annotations, for example, dialogue acts, topic segmentations, and meeting decisions. Figure 3.1 shows an example of meeting room setup at Edinburgh for the recording. We highlight the rich multimodal information available in conversational multiparty meetings.

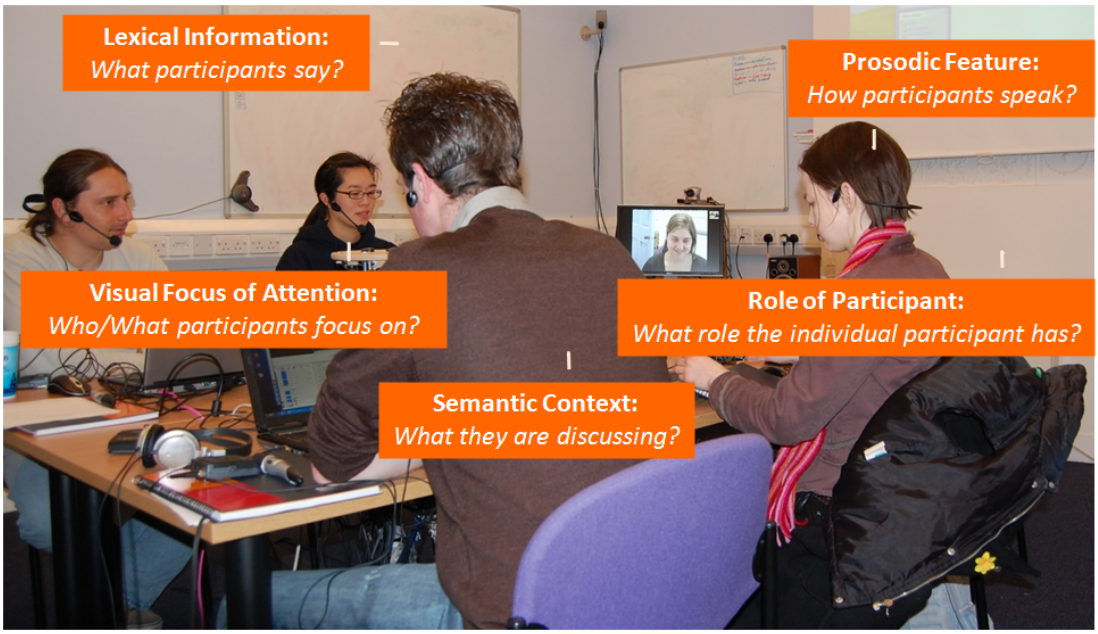


Figure 3.1: An example of meeting room setup for the AMI Meeting Corpus, in which we highlight the rich multimodal cues available in conversational multiparty meetings.

[8] http://corpus.amiproject.org

3.1.2 ICSI Meeting Corpus

The second meeting corpus used in Chapter 7 is the ICSI meeting corpus [Janin et al., 2003]. It consists of 75 non-scenario natural meetings, each approximately has one hour in length. The ICSI meeting corpus was recorded mainly by the ICSI researchers themselves, and the topics tend to be specialized and technical, e.g., about the discussion of speech and language technology.

Each meeting in the ICSI meeting corpus may have different number of both native and non-native participants. The speech is natural and spontaneous. Unlike the AMI corpus, which is multi-modal and contains a variety of information such as slides, whiteboard events and participant notes, the ICSI corpus consists entirely of speech and relevant annotations.

3.2 Multimodal Cues in Multiparty Meetings

We are concerned with LMs for multiparty meetings. The multimodal and interactive nature of group meetings demands a more comprehensive modelling framework for LMs to accommodate multimodal cues other than lexical information in LMs. Conventional n-gram models can be considered as a *flat* model since they rely on short-span lexical context. There are additional cues available in multiparty meetings, such as prosodic features, semantic context, participant roles and visual focus of attention. Our intuition is that these multimodal cues may augment the lexical context in an n-gram LM, and consequently be helpful for predicting the next word in an LM. This leads to our proposal of a *structured* multimodal language model for meetings. However, there are several difficulties to be overcome. Firstly, multimodal cues in meetings are normally heterogeneous, with different types and different scales. This makes the modelling problem difficult, because directly using MLE-based n-gram models for this task tends to overfit and make data sparsity more acute. Secondly, we require unsupervised methods to automatically extract some multimodal cues such as semantic context. These factors motivate us to investigate a novel approach based on a Bayesian framework.

3.2.1 Lexical Information

Language models can not exist without lexical information. Traditional LMs are estimated over sequential word streams, in a sentence-by-sentence manner. However in

multiparty meetings, there is not a single, linear stream of words. Instead, lexical cues from other speakers might also provide useful information for word prediction. In this case, it is not always obvious what the closest history is. By saying lexical information here, we especially refer this to the lexical cues from other speakers rather than the speaker in question.

Ji and Bilmes introduced a multi-speaker language model (MSLM) to augment a normal trigram context in conversational environments [Ji and Bilmes, 2004]. The basic belief for this model is that words from one speaker may be significantly affected by the words from another speakers in multiparty conversations. In particular, MSLM considers words from other speakers besides words from the current one during predicting, based on factored language model (FLM) and general parallel backoff (GPB) [Bilmes and Kirchhoff, 2003; Kirchhoff et al., 2003]. In other words, the history $\mathbf{u}_i = (w_{i-n+1}, \ldots, w_{i-1})$ in an n-gram LM is augmented by a_i, the closest previous word said by another speaker, $\Phi(\mathbf{u}_i) = (w_{i-n+1}, \ldots, w_{i-1}, a_i)$. The augmented context word a_i is obtained by aligning the multiple word streams, and taking the closest overlapping word from another speaker. Experiments on Switchboard and ICSI Meeting Recorder corpus [Janin et al., 2003] showed around 10% perplexity reduction. However, we argue that, although some reductions in perplexity have been achieved by MSLM, this approach still leaves much room for improvement. Firstly, the lexical context from other speakers a_i, which was sololy based on overlapping portions, is sparse, and the reaction time seems too short to model the interactive activity from a psychological view. Secondly, the mapping $\Phi(\mathbf{u}_i)$ on contexts can be seen as a partition over the space formed by the original context $\mathbf{u}_i$. Therefore, only for the cases in which there are still sufficient counts $C(\mathbf{u}_i, a_i)$ accounting for the robust estimation of MSLM after the partition might MSLMs potentially outperform the traditional n-gram LMs $P(w_i|\mathbf{u}_i)$.

Except for the above MSLM, little work can be found on *explicitly* modelling parallel lexical information in language models. However, there are such clear-defined conversational contexts that could be used in LMs in Switchboard and multiparty meetings, i.e, two-speaker dialogues in Switchboard and dialogue pairs in multiparty meetings, that it is indeed natural to consider lexical information from another speakers in ASR for conversational corpus. One reason for the lack of work in this area could be the trade-off between effort costs and performance gains for such new LMs, because the standard LMs, such as the simple yet efficient n-gram LMs, are hard to improve on. Another could be computational considerations. It is typically not easy to incorporate

longer contexts in LMs while maintaining the feasibility of effective computations.

3.2.2 Semantic Context

A limitation of current LMs is their inability to model long-span histories, due to the data sparcity problem. Integrating semantic information into contexts is a possible weapon to fight this limitation, in that parameter estimation over long-span history is avoided by taking only the inferred semantic topics into consideration, while still preserving the superiority of extended history information.

Bellegarda [Bellegarda, 1998] proposed the use of latent semantic analysis (LSA) to extract the semantic information from longer contexts, and combined the distance-based measures for predictive words with traditional n-gram probabilities. Different from applications in information retrieval, LSA-based approaches to extract semantic information for LMs do not have a natural way to define the *document*. A compromise is to take each sentence in the LM training data as a document. Semantic information is not so well-modelled in this way. For meetings, this problem becomes more severe, as there are much more short utterances and backchannels during spontaneous conversations. As a consequence, a well-defined *document* unit is important for semantic modelling in meetings. Another issue arising from the LSA-based semantic modelling approach is the combination of two different types of measures, probabilities from n-grams and distances from LSA, into probabilistic framework.

More recently, *topics* have been considered as a natural representation for semantic information in document modelling. Topics for contexts in LMs are assigned by either manually annotated or data-driven ways. With these topic assignments, contexts for conventional n-gram can then be augmented by various modelling approaches, e.g., Wu and Khudanpur [2002] used the maximum entropy (ME) principle to build a topic-dependent language model $P(w_i|\Phi(\mathbf{u}_i)) = P(w_i|\mathbf{u}_i, t_i)$, where t_i is the topic for $\mathbf{u}_i$.

However, the automatic extraction of topics from documents is more attractive, and has given rise to a number of topic models as introduced in Chapter 2, such as probabilistic LSA (pLSA) [Hofmann, 1999], and latent Dirichlet allocation (LDA) [Blei et al., 2003]. These generative models are mainly based on the "bag-of-words" assumption, that is, the order of words in documents is ignored. We are interested in probabilistic topic models such as LDA , which are easier to incorporate with other probabilistic models such as n-gram LMs. Some previous work has been done in the area of combining n-gram models and topic models such as LDA and probabilistic

latent semantic analysis (pLSA) for ASR on different data, for example, broadcast news [Mrva and Woodland, 2006; Tam and Schultz, 2006], lecture recordings [Hsu and Glass, 2006], and Japanese meetings [Akita et al., 2007]. Wallach [2006] proposed a more tight combination of Bayesian n-gram models and LDA models within a hierarchical Bayesian framework. We will introduce our extension to [Wallach, 2006] in Chapter 6.

For conversational speech corpus like meetings, dialogue act (DA) annotations are specifically available for AMI and ICSI corpora, and can be thought of as a means of encoding some semantic information for utterances. DAs normally represent the function of utterances, taking the form of speech-act-like units such as 'Statement', 'Question', 'Backchannel', 'Agreement', and 'Disagreement'. Intuitively, different DA has its specific words, i.e., it is more probable to see words such as 'Yeah' or 'Um' in a backchannel DA. Therefore, DA can potentially serve as a promising cue to assist language modelling in different ways. As in [Taylor et al., 1998], DA-specific language models can be build for each type of DAs, and then combined by interpolation. Alternatively, DA information can be used to extend contexts for LMs to make LMs be aware of high level semantics, $P(w_i|\Phi(\mathbf{u}_i)) = P(w_i|\mathbf{u}_i, d_i)$ where d_i is DA type for the context $\mathbf{u}_i$.

3.2.3 Role of Participant

Role information has often been used in the field of social network modelling [McCallum et al., 2005]. Currently, it is a relatively new idea to used role information for ASR. The role of a participant has been considered as one potential cue for language modelling, partly due to the availability of *scenario* meeting data in AMI project[9]. In AMI meeting scenario, the four participants play the roles of employees in an electronics company that decides to develop a new prototype of television remote control. The structure of the design process consists of four phases, namely project kick-off, functional design, conceptual design, and detailed design, with one meeting per design phase. Within this scenario, each of the four participants in meetings is given a different role to play, i.e., project manager (PM), industrial designer (ID), marketing expert (ME), and user interface designer (UI). The intuition behind using role of participant cues for language modelling is that a participant with a definite role is more likely to speak those words that are more specific to his/her role.

[9] Available online at `http://www.idiap.ch/amicorpus/`

McCallum et al. proposed an Author-Recipient-Topic (ART) model [McCallum et al., 2005]. The ART model, where the distribution over topics is conditioned distinctly on both the sender and the recipient, is an extension to the LDA model [Blei et al., 2003]. Since the ART model discovers the topics according to the relationships between people, it can be used to predict the role-equivalence between them, i.e., in a email corpus. The ART model has also been augmented as the Role-Author-Recipient-Topic (RART) model [McCallum et al., 2005], where authors, roles, and topics are modelled simultaneously. The RART can discover automatically person-role information by its explicit inclusion of role variables for both the sender and the recipient.

Participant role has been actively exploited within the AMI/AMIDA project consortium. Examples include using role of participant information in applications such as speech summarization [Murray et al., 2005], DA classification [Dielmann and Renals, 2008], and meeting decision detection [Hsueh and Moore, 2007].

3.2.4 Suprasegmental Feature

Suprasegmental features are prosodic features of a unit of speech, such as a syllable, word, phrase, or clause, which affect all the segments of the unit[10]. Typical prosodic features include pitch, tone, lexical stress, intonation, and rhythm in speech.

Prosodic features have long been studied as a plausible and complementary knowledge source for speech understanding, and have been successfully incorporated into speech systems in a variety of ways, such as topic segmentation [Shriberg et al., 2000], disfluency detection [Shriberg et al., 1997, 2001], speaker verification [Sönmez et al., 1998], and speech recognition [Stolcke et al., 1999; Chen and Hasegawa-Johnson, 2003; Chan and Togneri, 2006].

In fact there is a great deal of work on using prosodic features in speech and language processing. We thus focus our attention on the meeting domain. Conversational speech in multiparty meetings is more natural and spontaneous than read or acted speech. The prosodic behaviours for speech in meetings is therefore much less regular. Can prosody aid the automatic processing of multiparty meetings? Shriberg et al. [2001] gave the answer 'yes' to this question, from the evidence of successfully exploiting prosodic features for predicting punctuation, disfluency, and overlapping in meetings. It has also been noted that prosodic features can serve as an effi-

[10]The definition is derived from `http://en.wikipedia.org/wiki/Suprasegmental`.

cient non-lexical feature stream for various tasks on meeting corpora, such as dialogue acts segmentation and classification [Dielmann and Renals, 2008], speech summarization [Murray et al., 2005], and topic segmentation and classification [Hsueh and Moore, 2006].

The acoustic model, the language model, and the lexicon are the three essential components in a state-of-the-art ASR system. Prosody can be incorporated into all three levels. A basic approach to incorporate prosodic features in acoustic models for ASR uses "early integration", in which the prosodic features are appended to the standard acoustic features [Lei et al., 2006]. Early work to utilize prosody in language models used prosodic features to evaluate possible parses for recognized words, which in turn would be the basis for reordering word hypotheses [Veilleux and Ostendorf, 1993]. More recently, approaches that integrate prosodic features with LMs have emerged, in which LMs are conditioned on prosodic evidence by introducing intermediate categories. Taylor et al. Taylor et al. [1998] took the dialogue act types of utterances as the intermediate level, by first using prosodic cues to predict the DA type for an utterance and then using a DA-specific LM to constrain word recognition. Stolcke et al. Stolcke et al. [1999] instead used prosodic cues to predict the hidden event types (filled pause, repetition, deletion, repair) at each word boundary using a hidden event n-gram model, and then conditioned the word portion of the n-gram on those hidden events. Chan and Togneri [2006] proposed to incorporate prosody into LMs using maximum entropy. However, the prosodic features they used were derived from manual ToBI transcriptions. An example of using prosody in the lexicon was provided by Chen and Hasegawa-Johnson [2003], where prosodic features, such as stress and phrase boundary, were included in the vocabulary. Each word had different variations corresponding to stress and whether or not it precede a prosodic phrase boundary. This approach attempted to capture the effects of how prosodic features affect the spectral properties of the speech signal and the co-occurrence statistics of words.

Most research on using prosodic features for ASR has been applied to small and task-oriented databases. The goal of effectively using prosody for large-vocabulary speech recognition, such as recognition of meeting speech, still remains elusive. There has been little work in this direction in the meeting domain. One reason for this is due to the difficulty of modelling the relationship between symbolic words and normally non-symbolic prosodic features. From the above examples, we can see that one question during the application of prosody is how to define the intermediate categories between prosodic features and target categories (i.e., dialogue acts, phrase boundaries,

and words)? For tasks such as disfluency detection and dialogue act classification, there are many fewer target categories, i.e., several tens or hundreds. For ASR task, however, the number of word categories (vocabulary) is generally more than tens of thousands, which results in the failure of discriminabilities of words in view of prosodic features. Therefore, to find an approximate prosodic representation for each word vocabulary is one way to use prosodic features for ASR.

Since it is difficult to modelling prosody via intermediate representations, Shriberg and Stolcke [2004] proposed direct modelling of prosodic features. In this approach, prosodic features F are extracted directly from the speech signal. Machine learning techniques (e.g., Gaussian Mixture Model, decision tree) then determine a statistical model $P(C|F)$ to use prosodic features in predicting the target classes of interest C. No human annotation of prosodic events is required in this case. However, we argue that the problem still implicitly exists, because using prosodic features to predict a very large number of target categories like words will again fail in capturing the prosodic discriminabilities.

It is well accepted that humans are able to understand prosodic structure without lexical cues. Sub-lexical prosodic analysis [Aylett, 2006] attempts to mimic this human ability using syllable finding algorithms based on band pass energy. Prosodic features are then extracted at the syllable level. The extraction of syllable-based prosodic features is attractive, because the syllable is accepted as a means of structuring prosodic information. This approach was verified on DA and hotspot categorization [Aylett, 2006], which encourages us to utilize syllable-based prosodic features in LMs for ASR.

3.2.5 Other Cues

Though people usually communicate by speech, communication is in fact a multimodal process [Krauss et al., 1977]. A multimodal recognition architecture is desirable as it affords an opportunity to resolve ambiguity, and various architectures are emerging which integrate semantically rich modalities with this aim [Oviatt, 2002]. Communicative success arises with the inclusion of multimodal cues because in that case interlocutors can then make use of other types of information other than just the speech signal. Linguistic utterances are inevitably ambiguous. Non-verbal multimodal cues, such as eye gaze, gesture, hand and head motion, provide complementary information to lexical cue, and can thus potentially resolve ambiguity.

The modality of eye gaze has been used together with speech for human-computer

interaction in recent decades [Koons et al., 1993; Campana et al., 2001; Zhang et al., 2003; Cooke and Russell, 2005]. As noted by [Russell et al., 2004], the speaker's eye gaze during a communication is typically ahead of the speech, while the listener's eye gaze by contrast usually follows the speech signal of speakers. Although it is still unknown what the consequences are for the listener's speech processing caused by the asynchrony between speakers and listeners looking at same objects, it is interesting to investigate the effects of speakers' leading eye gaze before their speech signal, and the relationship between these two modalities. In [Koons et al., 1993; Campana et al., 2001], simultaneous error-free inputs from speech, gaze and gesture from users were combined, demonstrating the mutual information from different modalities. In contrast to error-free inputs, Zhang et al. [2003] proposed using eye tracking information to recover errors of speech recognition due to misnaming objects. The goal is to find how to effectively make use of the error-prone information from both speech and gaze modalities, and use one modality to correct errors from the other. In their experiments, the task of participants was to name objects, which had phonologically similar sounds, on a visual display unit (VDU), and the nearest object to the eye was used to resolve which object was being referred to. Cooke and Russell [2005] attempted to use eye gazes in their more natural role as a passive modality, and extended applications to natural dialogues with spontaneous speech. They developed a bimodal ASR using visual focus of attention (FOA) as a secondary modality to drive a dynamic language model. The scenario was map description task involving two participants, with an instruction 'giver' describing a map, containing objects and routes around them, to an instruction 'follower'. Prior knowledge of which words were associated with which objects nearest to eye gaze on the map were used to reorder N-Best lists from ASR recognizer. Only marginal improvement in WER was observed. Instead of using eye gaze, Roy and Mukherjee [2005] integrated visual scene context into the speech recognition and understanding process in an object selection task where the system finds the object in a scene that best fits the meaning of a spoken description. Visual scene context is used to dynamically modify the language model probabilities on the basis of visual context, in a visually-driven semantic priming form.

The above example research exploited visual modality (i.e., eye gaze, visual scene context) mostly in command driven interfaces or task-constrained situations. For the case of real data, i.e., multiparty meetings, it is more challenging to integrate lexical and visual cues for language modelling due to the free and natural communication in meetings. We therefore propose to utilize a higher level visual cue - visual focus of

attention (VFOA). Attention refers to the cognitive process of selectively concentrating on one thing while ignoring other things, which plays an important role in human face-to-face communications. In the AMI meeting corpus, there are some manually annotated VFOA information, which indicate the object/participant at which the speaker is gazing such as the table, whiteboard, slidescreen, and all other relevant meeting participants. The VFOA information in meetings is anticipated to provide knowledge of the interaction between participants in meeting dynamics, and consequently be used to extend the contexts for predicting in LMs based on the longer-span contexts.

3.3 AMI-ASR Baseline System for Meetings

The baseline ASR system we used in the experiments in this thesis was developed by the AMI-ASR team [Hain et al., 2007]. The AMI-ASR system uses the standard ASR framework, i.e., hidden Markov model (HMM) based acoustic modelling and *n*-gram language models. Figure 3.2 shows the framework of multiple-pass processing of the AMI-ASR system. We will briefly describe the main components of the AMI-ASR system. Refer to [Hain et al., 2007] for details.

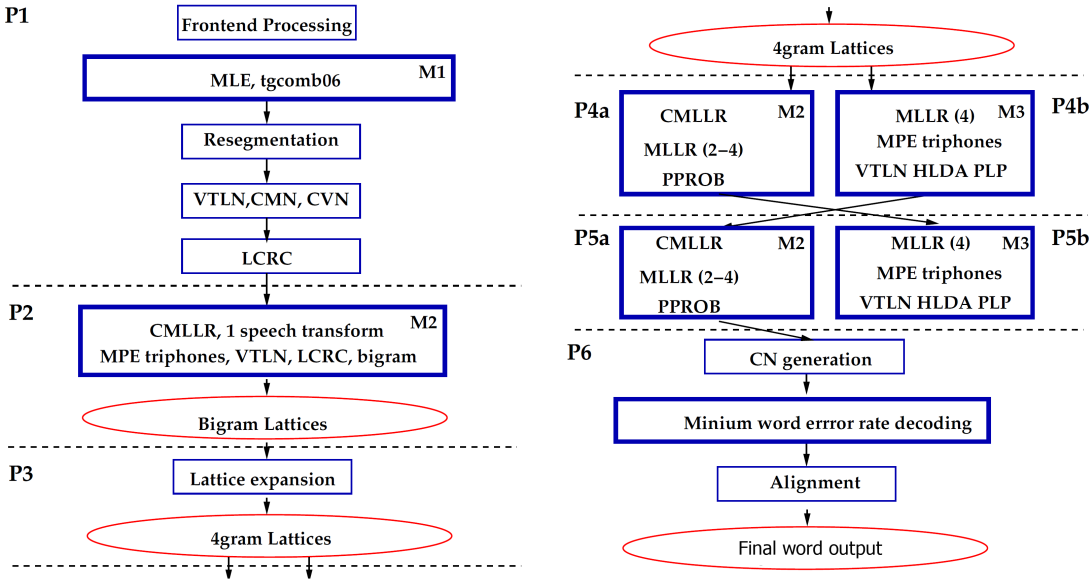


Figure 3.2: The framework of the AMI-ASR system, which serves as the baseline of all the experiments in this thesis. This figure was derived from [Hain et al., 2007].

We only consider the processing of audio recorded from individual head microphones (IHM). As observed in [Hain et al., 2007], there is a substantial performance degradation for multiple distant microphones (MDM) data comparing to IHM data.

In the frontend processing step, the audio is segmented, normalized for input channels, and suppressed for noise and cross-talk. Feature streams, with vectors consisting of 12 MF-PLP features and raw log energy and first and second order derivatives, are extracted from the audio signals. Cepstral mean and variance normalization (CMN/CVN) is further performed on a per channel basis.

HMM-based acoustic models use cross-word state-clustered triphone models. Various modelling techniques have been applied and produced consistent performance improvements, for example, vocal tract length normalization (VTLN), heteroscedastic linear discrimant analysis (HLDA), discriminative training using the minimum phone error (MPE), and speaker adaptive training (SAT).

The AMI-ASR system uses standard n-gram LMs and builds LMs using the SRILM toolkit [Stolcke, 2002]. Special techniques are used to collect from the Web text data that is different from the already existing background material [Wan and Hain, 2006]. Interpolated bigram, trigram, and 4-gram LMs were trained using 10 text sources including AMI, Fisher, Hub4, ICSI, ISL, NIST, Switchboard, and three web-data. The lexicon is based on the UNISYN pronunciation lexicon[11], and consists of 50k words.

There are in total six passes in the AMI-ASR system, as shown in Figure 3.2. For experiments in this thesis, we generally take the acoustic models, *M2*, from the second pass, replace the language model as our new one, and do the decoding. The *M2* acoustic models are trained using VTLN, HLDA, SAT, MPE and phoneme-state posteriors LCRC features. They are adapted using the transcription from the first pass and applied a single constrained maximum likelihood regression (CMLLR) transform. The WER of the second pass using an interpolated bigram LM trained on the 10 text resources mentioned above, as reported in [Hain et al., 2007], is 29.2% on *rt06seval* IHM test set. A rescoring using an interpolated 4-gram LM on the bigram lattices yields an WER of 26.0%. Since we generally use an trigram LM without interpolation in the experiments of this thesis, we normally obtain an WER between 29.2% and 26.0% on *rt06seval*.

3.4 Summary

In this chapter, we reviewed some previous work on incorporating the multimodal cues from multiparty conversational meetings into language models. This is complementary to Section 2.2.3.2 on using non-linguistic knowledge for language modelling. Multimodal cues described in this chapter are mainly specific to the meeting domain. We

[11] http://www.cstr.ed.ac.uk/projects/unisyn/

also introduced the AMI-ASR baseline system for meetings, which provides the experimental basis for experiments and results in the rest of this thesis.

Chapter 4

Bayesian Language Modelling for ASR using the Pitman-Yor Process

In this chapter we investigate the application of a hierarchical Bayesian language model (LM) based on a nonparametric prior called the Pitman-Yor process for automatic speech recognition (ASR) of multiparty meetings. The hierarchical Pitman-Yor language model (HPYLM) provides a Bayesian interpretation of LM smoothing. An approximation to the HPYLM recovers the exact formulation of the interpolated Kneser-Ney smoothing method in n-gram models [Teh, 2006b]. The HPYLM offers a principled approach to language model smoothing, embedding the power-law distribution for natural language. This chapter focuses on the application and scalability of HPYLM on a practical large vocabulary ASR system. Experimental results on NIST RT06s evaluation meeting data verify that HPYLM is a competitive and promising language modelling technique, which consistently performs better than interpolated Kneser-Ney and modified Kneser-Ney n-gram LMs in terms of both perplexity and word error rate.

4.1 Introduction

The main goal of this chapter is to carry out a comprehensive study on the application of the HPYLM to large vocabulary ASR. The HPYLM is a theoretically elegant language model first proposed in the machine learning field [Teh, 2006b]. In the speech community, however, two questions remain interesting and have not been studied before. First, will the HPYLM work for ASR tasks, which are normally evaluated in terms of word error rate (WER)? Second, is it possible to scale up the HPYLMs to work

on large vocabulary ASR using large training corpora? In the rest of this chapter, we provide our answers to these two questions, by extending our previous work in [Huang and Renals, 2007a] including the presentation of a parallel training algorithm, more detailed descriptions, more experimental results, and a thorough discussion.

For the first question, we verify the HPYLM in terms of both perplexity and WER using an efficient computational implementation. In recent years there has been a growing research interest in the automatic transcription of multiparty meetings (see detailed introduction in Chapter 3), which is typically one of the first several essential steps for further processing of meetings, such as information retrieval and summarization. European projects AMI and AMIDA [Renals et al., 2007] are examples of such efforts. This has provided us a well-defined benchmark on which to evaluate a state-of-the-art large vocabulary ASR system. We present comprehensive experimental results on multiparty conversational meeting corpora, and observe consistent and significant reductions in perplexity and WER in comparison to the interpolated Kneser-Ney language model (IKNLM) [Kneser and Ney, 1995] and the modified Kneser-Ney language model (MKNLM) [Chen and Goodman, 1999], which are the state-of-the-art smoothing methods for language modelling.

It is often expensive to do Bayesian inference on large training data. In order to obtain a sufficiently large language model for state-of-the-art large vocabulary ASR systems, we have developed a parallel algorithm for the estimation of an HPYLM, enabling the use of a large training corpus. We also demonstrate that inference of an HPYLM converges quickly, taking only a few tens of iterations to converge to a language model of comparable (or better) accuracy than the IKNLM or the MKNLM.

4.2 Hierarchical Bayesian Language Models based on Pitman-Yor Processes

We introduce a Bayesian language model based on Pitman-Yor processes using a hierarchical framework. This section briefly summarizes the original work on the HPYLM [Teh, 2006b; Goldwater et al., 2006b], and refers to our published work in [Huang and Renals, 2007a, 2010b].

4.2.1 Hierarchical Pitman-Yor Process Language Models

The Pitman-Yor process can be used to create a two-stage language modelling framework [Goldwater et al., 2006b]. Following the Chinese restaurant metaphor discussed in section 2.3.1.3, a language model can be viewed as a restaurant in which each table has a label of a word w generated by $G_b(w)$, and multiple tables can have the same label. Each customer represents a word token, so that the number of customers at a table corresponds to the frequency of the lexical word labeling that table. A customer may only be assigned to a table whose label matches that word token.

Consider a vocabulary $\mathcal{V}$ with $|\mathcal{V}|$ word types. Let $G_{\emptyset}(w)$ be the unigram probability of w, and $G_{\emptyset} = [G_{\emptyset}(w)]_{w\in\mathcal{V}} = [G_{\emptyset}(w_1), G_{\emptyset}(w_2), G_{\emptyset}(w_3), \ldots, G_{\emptyset}(w_{|\mathcal{V}|})]$ represents the vector of word probability estimates for unigrams. A Pitman-Yor process prior, as described in Section 2.3.1.3, is placed over $G_{\emptyset} \sim \mathrm{PY}(d_0, \theta_0, G_b)$ with an uninformative base distribution $G_b(w) = 1/|\mathcal{V}|$ for all $w \in \mathcal{V}$. According to the Chinese restaurant metaphor, customers (word tokens) enter the restaurant and seat themselves at either an occupied table or a new one, with probabilities expressed in Equation (2.27). Each table has a label $w \in \mathcal{V}$ initialized by the first customer seated on it, and the next customer can only sit on those tables with the same label. Those c_w customers that correspond to the same word label w, can sit at different tables, with t_w denoting the number of tables with labels w. Given the seating arrangement $\mathcal{S}$ of customers, and the hyperparameters d_0 and θ_0, the predictive probability of a new word w in a unigram model is given in Equation (4.1), by collecting probabilities in Equation (2.27) corresponding to each label w for tables:

$$\begin{aligned} P(w|\mathcal{S}, d_0, \theta_0) &= \sum_{k=1}^{t_\bullet} \frac{c_k - d_0}{\theta_0 + c_\bullet}\delta_{kw} + \frac{\theta_0 + d_0 t_\bullet}{\theta_0 + c_\bullet} G_b(w) \\ &= \frac{c_w - d_0 t_w}{\theta_0 + c_\bullet} + \frac{\theta_0 + d_0 t_\bullet}{\theta_0 + c_\bullet} G_b(w) \end{aligned} \tag{4.1}$$

where δ_{kw} equals to 1 if table k has the label of w, and 0 otherwise, $c_\bullet = \sum_w c_w$ is the total number of customers, and $t_\bullet = \sum_w t_w$ is the total number of tables, in the restaurant for unigrams. Averaging over seating arrangements and hyperparameters, we can obtain the probability $P(w)$ for a unigram LM. When $d_0 = 0$, the Pitman-Yor process reduces to a Dirichlet distribution $\mathrm{Dir}(\theta_0 G_b)$, and Equation (4.1) becomes the predictive distribution in a Bayesian language model using the Dirichlet distribution as the prior [MacKay and Peto, 1994]:

$$P(w|\mathcal{S}, \theta_0) = \frac{c_w}{\theta_0 + c_\bullet} + \frac{\theta_0 G_b(w)}{\theta_0 + c_\bullet} = \frac{c_w + \theta_0 G_b(w)}{c_\bullet + \theta_0}. \tag{4.2}$$

This can be regarded as an additive smoothing of the empirical probability ($c_w/c_\bullet$), by balancing the empirical counts (c_w) with the additive pseudo-counts ($\theta_0 G_b(w)$) of the prior Dirichlet distribution.

We can generalize the above unigram example to the n-gram case. An n-gram LM defines a probability distribution over the current word given a context $\mathbf{u}$ consisting of $n-1$ words. Let $G_{\mathbf{u}}(w)$ be the probability of the current word w and $G_{\mathbf{u}} = [G_{\mathbf{u}}(w)]_{w\in\mathcal{V}}$ be the target probability distribution given the context $\mathbf{u}$. A Pitman-Yor process is served as the prior over $G_{\mathbf{u}}$, with discounting parameter $d_{|\mathbf{u}|}$ and strength parameter $\theta_{|\mathbf{u}|}$ specific to the length of the context $|\mathbf{u}|$. The base distribution is $G_{\pi(\mathbf{u})}$, the lower order model of probabilities of the current word given all but the earliest word in the context. That is,

$$G_{\mathbf{u}} \sim \mathrm{PY}(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \tag{4.3}$$

Since $G_{\pi(\mathbf{u})}$ is still an unknown probability distribution, a Pitman-Yor process is recursively placed over it with parameters specific to $|\pi(\mathbf{u})|$, $G_{\pi(\mathbf{u})} \sim \mathrm{PY}(d_{|\pi(\mathbf{u})|}, \theta_{|\pi(\mathbf{u})|}, G_{\pi(\pi(\mathbf{u}))})$. This is repeated until we reach $G_\emptyset$ for a unigram model discussed above. This results in a hierarchical prior (Figure 4.1), enabling us to generalize from the unigram to the n-gram case. There are multiple restaurants (Pitman-Yor processes) in the prior hierarchy, with each corresponding to one context. By using the hierarchical framework of Pitman-Yor priors, different orders of n-gram can thus share information with each other, similar to the traditional interpolation of higher order n-grams with lower order n-grams.

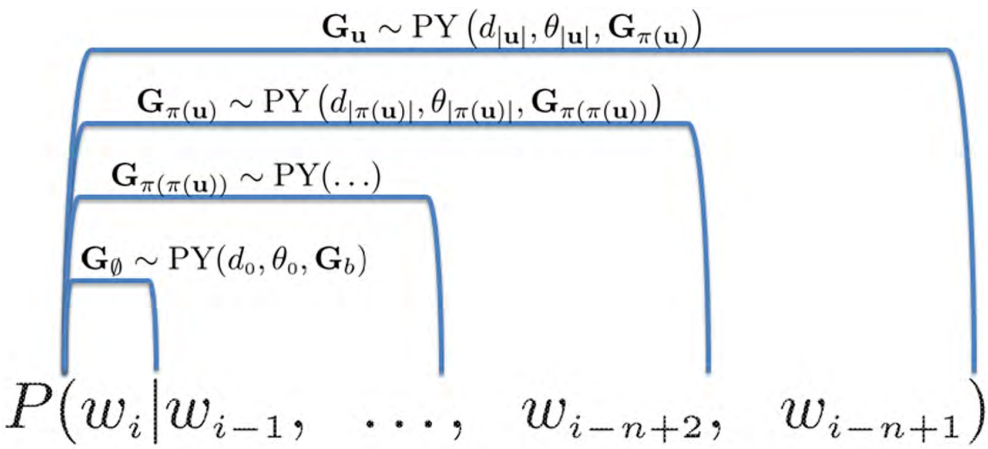


Figure 4.1: The hierarchy of Pitman-Yor process priors for n-gram LMs. Pitman-Yor processes are placed recursively as priors over the n-order predictive distributions until we reach the unigram model $G_\emptyset$. $\pi(\mathbf{u})$ denotes the back-off context of $\mathbf{u}$.

Based on this overall framework for an HPYLM, a central task is the inference of seating arrangements in each restaurant and the estimation of the context-specific parameters from the training data. Given training data $\mathcal{D}$, we know the number of co-occurrences of a word w after a context $\mathbf{u}$ of length $n-1$, $c_{\mathbf{u}w}$. This is the only information we need to train an HPYLM. An MCMC algorithm can be used to infer the posterior distribution of seating arrangements. We use Gibbs sampling to keep track of which table each customer sits at, by iterating over all customers present in each restaurant — first removing a customer w from the restaurant $\mathbf{u}$, and then adding the customer w back to the restaurant $\mathbf{u}$ by resampling the table at which that customer sits. After a sufficient number of iterations, the states of variables of interest in the seating arrangements will converge to the required samples from the posterior distribution. In the HPYLM the more frequent a word token, the more likely it is there are more tables corresponding to that word token.

For an n-gram LM, there are $2n$ parameters $\Theta = \{d_m, \theta_m : 0 \leq m \leq n-1\}$ to be estimated in total. We use a sampling method based on auxiliary variables as in [Teh, 2006a], which assumes that each discount parameter d_m has a prior beta distribution with auxiliary variables a_m and b_m while each concentration parameter θ_m has a prior gamma distribution with auxiliary variables α_m and β_m.

Under a particular setting of seating arrangements $\mathcal{S}$ and hyperparameters Θ, the predictive probability $P(w|\mathbf{u}, \mathcal{S}, \Theta)$ can be obtained similarly to the case for unigram in Equation (4.1) for each context $\mathbf{u}$:

$$\begin{aligned} P(w|\mathbf{u}, \mathcal{S}, \Theta) &= \frac{c_{\mathbf{u}w\bullet} - d_{|\mathbf{u}|} t_{\mathbf{u}w}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\bullet\bullet}} \\ &\quad + \frac{\theta_{|\mathbf{u}|} + d_{|\mathbf{u}|} t_{\mathbf{u}\bullet}}{\theta_{|\mathbf{u}|} + c_{\mathbf{u}\bullet\bullet}} P(w|\pi(\mathbf{u}), \mathcal{S}, \Theta) \end{aligned} \tag{4.4}$$

in which if we set the discounting parameters $d_{|\mathbf{u}|} = 0$ for all $\mathbf{u}$, we resort to a hierarchical Dirichlet language model (HDLM) [MacKay and Peto, 1994], similar to Equation (4.2). The HDLM and the HPYLM share the same idea of interpolation with the lower order n-grams. The difference is that the HPYLM explores discounts from empirical counts, while the HDLM does not.

The overall predictive probability can be approximately obtained by collecting I samples from the posterior over $\mathcal{S}$ and Θ, and then averaging Equation (4.4) to approx-

imate the integral with samples:

$$P(w|\mathbf{u}) \approx \sum_{i=1}^{I} P(w|\mathbf{u}, \mathcal{S}^{(i)}, \Theta^{(i)})/I \ . \tag{4.5}$$

If we assume that the strength parameters $\theta_{|\mathbf{u}|} = 0$ for all $\mathbf{u}$, and restrict $t_{\mathbf{u}w}$ to be at most 1 (i.e., all customers representing the same word token should only sit on the same table together), then the predictive probability in Equation (4.4) directly reduces to the predictive probability given by the IKNLM in Equation (2.15). We can thus interpret IKN as an approximate inference scheme for the hierarchical Pitman-Yor process language model [Teh, 2006b].

4.2.2 Inference

In the HPYLM, we are interested in the posterior distribution over the latent predictive probabilities $\mathcal{G} = \{G_{\mathbf{u}} : \text{all contexts } \mathbf{u}\}$ and the hyperparameters $\Theta = \{d_m, \theta_m : 0 \leq m \leq n-1\}$, given the training data $\mathcal{D}$. The hierarchical Chinese restaurant process represents it as the seating arrangement, denoted by $\mathcal{S} = \{S_u : \text{all contexts } \mathbf{u}\}$, in the corresponding restaurant, by marginalizing out each $G_{\mathbf{u}}$. The central task for the inference is thus to infer the posterior distribution over the seating arrangements $\mathcal{S} = \{S_u : \text{all contexts } \mathbf{u}\}$ and the hyperparameters $\Theta = \{d_m, \theta_m : 0 \leq m \leq n-1\}$ given the training data $\mathcal{D}$: $P(\mathcal{S}, \Theta|\mathcal{D}) = P(\mathcal{S}, \Theta, \mathcal{D})/P(\mathcal{D})$. With the posterior, we can calculate the predictive probabilities according to Equation (4.4), and Equation (4.5) by further integrating out $\mathcal{S}$ and Θ. We follow inference schemes based on MCMC for the HPYLM [Teh, 2006a,b], depicted in Algorithm 1.

For ADDCUSTOMER($\mathbf{u}, w$), we use Equation (2.27) to sample a new seating table in restaurant $\mathbf{u}$ for customer w. For REMOVECUSTOMER($\mathbf{u}, w$), we simply remove a customer from k^{th} table according to the probability proportional to the number of customers already seated there $c_{\mathbf{u}wk}$. For SAMPLEPARAMS(m), a sampling method based on auxiliary variables is used for sampling the hyperparameters d_m and θ_m, which assumes that each discount parameter d_m has a prior beta distribution $d_m \sim \text{Beta}(a_m, b_m)$ while each strength parameter θ_m has a prior gamma distribution $\theta_m \sim \text{Gamma}(\alpha_m, \beta_m)$ [Teh, 2006a,b].

Algorithm 1: INFERHPYLM(n): an algorithm for the inference of an HPYLM, where n is the order of LMs, $\mathbf{u}$ is the context (restaurant), w is the word token (customer), n_w is the number of occurrences of w after context $\mathbf{u}$. d_m and θ_m are the discount and strength parameters of Pitman-Yor processes for n-grams of order m.

```
procedure INFERHPYLM( order n )
input : the order of n-gram with n > 0
1  for each order m = 0 to n − 1 do
2      for each context u of order m do
3          for each word w appearing after context u do
4              for i = 0 to n_w do iterations for n_w times
5                  if REMOVECUSTOMER( u, w ) then
6                      ADDCUSTOMER( u, w );
7                  endif
8              endfor
9          endfor
10     endfor
11     (d_m, θ_m) = SAMPLEPARAMS( m );        /* sampling hyperparameters
       d_m, θ_m for order m */
12 endfor
```

4.3 A Parallel Training Scheme for the HPYLM

Training an HPYLM typically uses Markov chain Monte Carlo (MCMC) sampling methods, which are significantly more computationally expensive than training an IKNLM or MKNLM. There are two things that lead to increased computational complexity when training an HPYLM. First, the inference itself can take several hundred sampling iterations to converge. Second, the memory requirement for training an HPYLM is large and grows linearly with corpus size. It is therefore important to make the HPYLM scale to work on the large corpora used in LVASR and SMT.

In this section, we present a parallel training algorithm for the HPYLM. The parallel training algorithm alleviates the limitation of computational time and memory constrained by a single machine, by dividing the inference into sub-tasks. The sub-tasks, thereafter, can be either parallelly submitted to a computing cluster, or sequentially run on a single server machine. In this way, we are able to train in parallel HPYLMs on

corpora of more than 200 hundred millions of words, which in turn reduce the perplexity and word error rate significantly on meeting transcription tasks. We show that any approximations resulting from the proposed parallel algorithm have a negligible effect on performance.

Even with an efficient implementation of the HPYLM, however, it is still computationally expensive in terms of computing time and memory requirements to infer an HPYLM using a large corpus. According to our previous results in [Huang and Renals, 2007a], increasing the size of either corpora or vocabulary increases the computational complexity of inferring an HPYLM, with a corpus of around 50 million words using a 50k vocabulary roughly taking ten minutes per iteration and occupying about 2.5 GB memory.

4.3.1 Algorithm

This motivated us to design a parallel training algorithm to efficiently estimate an HPYLM. We use a *divide-and-conquer* scheme. There are two steps: *data partition* and *model combination*. Generally speaking, we divide the inference task, which is normally infeasible or expensive using a single machine, into sub-tasks that fit well to the computational capacity of a single computing node – alleviating the memory requirement. Further combined with parallelism, we can also decrease the computational time for inference by running sub-tasks in parallel.

In the data partition step, we first divide word types in the vocabulary $\mathcal{V}$ into subsets $\mathcal{V}_k \subset \mathcal{V}$. For each subset $\mathcal{V}_k$, we then compose those bigrams beginning with words $w \in \mathcal{V}_k$, and their corresponding child n-grams with $n > 2$, as a sub-HPYLM (dotted rectangles), and put a pseudo $G_\emptyset$ (dotted circles) as the Pitman-Yor process for unigrams of the sub-HPYLM, as shown in Figure 4.2. Each sub-HPYLM can be inferred separately using the same routines as those for a normal HPYLM, except that the pseudo $G_\emptyset$ now additionally collects the number of insertion and deletion for customer $w \in \mathcal{V}_k$. The inference of sub-HPYLMs can be executed in parallel by submitting to a computing cluster.

In the model combination step, we combine all the sub-HPYLM models level-by-level in the HPY hierarchy. For each level, we accumulate auxiliary parameters, and sample the hyperparameters d and θ. For the global $G_\emptyset$ for unigrams, we infer the seating arrangements by using the insertion and deletion statistics accumulated by each pseudo $G_\emptyset$, to make sure the HPYLM is consistent regarding the *modified*

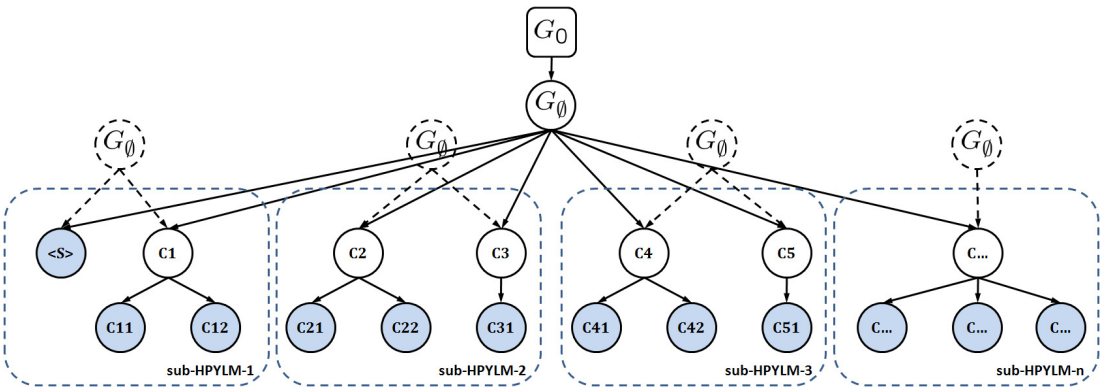


Figure 4.2: The partition of an HPYLM into sub-HPYLMs, denoted by dotted rectangles, for the parallel inference. The dotted circles represent pseudo $G_\emptyset$s to complete a Pitman-Yor process hierarchy, and collect additional insertion/deletion information. Each circle corresponds to a context, or a restaurant in the Chinese restaurant metaphor.

counts for lower order n-grams [Kneser and Ney, 1995; Teh, 2006b]. Depending on when to combine sub-HPYLMs, we explore two different versions of parallel training algorithm. The first version, iterative-synchronized or IterSync, combines sub-HPYLMs, and sample hyperparameters after each iteration, while the second one, final-synchronized or FinalSync, does the combination and samples hyperparameters only after each sub-HPYLM has finished all the predefined number of iterations. Due to the extra costs of submitting and queueing jobs at each iteration, it is much slower for IterSync to infer an HPYLM than FinalSync. It is understandable, however, that the second version, FinalSync, will lose more precisions than the first one, since the hyperparameters are not optimized globally, which in turn does harm to the inference of seating arrangements.

The parallel training algorithm makes it possible to estimate an HPYLM on large corpora using a large vocabulary. The detailed parallel algorithm for the HPYLM is described in Algorithm 2 for the IterSync version and Algorithm 3 for the FinalSync version. Since this is indeed data parallelism, it is possible to port the parallel training algorithm for the HPYLM in Algorithm 2 and Algorithm 3 to the MapReduce framework. For example, we can infer sub-HPYLMs (those lines between PARALLEL and ENDPARALLEL in Algorithm 2/ 3) in the Map step. After that, we then infer HPYLM for unigrams, collect statistics, and sample hyperparameters in the Reduce step.

Algorithm 2: InterSync - A Parallel Training Algorithm for the HPYLM, in which the function INFERHPYLM(n) was defined in Algorithm 1.

```
procedure PARALLELHPYLMITERSYNC(order n, u, V)
input : order n, counts u, and vocabulary V
divide vocab V into K subsets, V_k ⊂ V
for k = 1 to K do
    read counts for subset V_k from count u
    initialize sub-HPYLM_k
endfor
for i = 1 to N iterations do
    PARALLEL
    for k = 1 to K do
        infer sub-HPYLM_k: INFERHPYLM(n)
    endfor
    ENDPARALLEL
    infer HPYLM for unigrams
    collect statistics
    sample hyperparameters
endfor
combine sub-HPYLMs
estimate a final ARPA format LM
```

4.4 Implementation

We implemented the hierarchical Pitman-Yor process language model by extending the SRILM toolkit [Stolcke, 2002]. We highlight four characteristics of this implementation. First, it is consistent and coherent with the existing SRILM software. We inherited the HPYLM classes from the base SRILM classes, and provided the same interfaces for language modelling. Second, it has efficient memory management and computational performance by directly using the data structures available in SRILM. Third, it is a flexible framework for Bayesian language modelling. We can, for example, train a language model with Kneser-Ney smoothing for unigrams, modified Kneser-Ney smoothing for bigrams, and Pitman-Yor process smoothing for trigrams. Finally, this implementation is extensible for future developments: e.g., taking into accounts the combination with multimodal cues for language models via probabilistic

Algorithm 3: FinalSync - Another Parallel Training Algorithm for the HPYLM, in which the function INFERHPYLM(n) was defined in Algorithm 1.

```
procedure PARALLELHPYLMFINALSYNC(order n, u, V)
input : order n, counts u, and vocabulary V
divide vocab V into K subsets, V_k ⊂ V
for k = 1 to K do
    read counts for subset V_k from count u
    initialize sub-HPYLM_k
endfor
PARALLEL
for i = 1 to N iterations do
    for k = 1 to K do
        infer sub-HPYLM_k: INFERHPYLM(n)
    endfor
endfor
ENDPARALLEL
infer HPYLM for unigrams
collect statistics
sample hyperparameters
combine sub-HPYLMs
estimate a final ARPA format LM
```

topic models. There is a brief introduction to the executable program (hpylm) for the HPYLM in Section E.

We also implemented the parallel training algorithm, described in Section 4.3, for the HPYLM. The parallel computing scheme makes it possible to estimate an HPYLM using a large vocabulary on a large corpus. This work on parallel computing for inferring HPYLMs has made use of the computing cluster managed by Sun Grid Engine, which is provided by the Edinburgh Compute and Data Facility (ECDF)[1].

This implementation of an HPYLM outputs a standard ARPA format LM, with an identical format to a conventional n-gram LM. This makes it easy to evaluate the HPYLM in a conventional ASR system.

[1] http://www.ecdf.ed.ac.uk/

4.5 Experiments and Results

The experiments reported in this section were performed using the U.S. NIST Rich Transcription (RT) 2006 spring meeting recognition evaluation data (RT06s)[2]. We tested only on those audio data recorded from individual headset microphones (IHM), consisting of meeting data collected by the AMI project, Carnegie Mellon University (CMU), NIST, and VT (Virginia Tech).

The training data sets for language models used in this section, and the test transcription, are listed in Table 4.1. The web-data for meetings and conversational speech were collected from the world wide web using strategies described in [Wan and Hain, 2006]. We exploited different combinations of these training data sets for the following experiments.

Table 4.1: Statistics of the training and testing data sets for language models in the RT06s task.

No.	LM Data Set	#Sentences	#Words
1	AMI data from rt05s	68,806	801,710
2	ICSI meeting corpus	79,307	650,171
3	ISL meeting corpus	17,854	119,093
4	NIST meeting corpus-2	21,840	156,238
5	NIST meeting corpus-a	18,007	119,989
	MTRAIN (Sets 1–5)	205,814	1,847,201
6	FISHER (fisher-03-p1)	1,076,063	10,593,403
	MTRAIN + F (Sets 1–6)	1,281,877	12,440,604
7	WEBMEET (webdata meetings)	3,218,066	36,073,718
	ALL - WC (Sets 1–7)	4,499,943	48,514,322
8	WEBCONV (webdata conversational)	12,684,025	162,913,566
	ALL	17,183,968	211,427,888
test	*rt06seval*	3,597	31,810

[2] http://www.nist.gov/speech/tests/rt/

A second corpus we consider in this section is a domain-specific meeting corpus — the AMI Meeting Corpus[3] [Carletta, 2007], as described in Section 3.1.1. We used the scenario part of the AMI Meeting Corpus for our experiments, in a 5-fold cross validation setup. There are 137 scenario meetings in total, as shown in Table 4.2.

Table 4.2: Statistics of the 5-fold cross validation setup of the AMI scenario meetings.

FOLD	#Sentences	#Words	#Meetings
1	17,368	175,302	32
2	13,512	135,415	28
3	12,220	129,212	24
4	12,849	136,071	28
5	11,975	113,969	25
all	67,924	689,969	137

Most of the following experiments used a common vocabulary with $50,000$ word types, unless explicitly indicated otherwise. For the 5-fold cross validation experiments on the scenario AMI meetings, the vocabulary was slightly tuned for each *training* set, while keeping the vocabulary size fixed to $50,000$ word types. These vocabularies were those used in the AMI-ASR system [Hain et al., 2007].

The lower discounting cutoffs of the n-gram counts (i.e., *-gt1min*, *-gt2min*, and *-gt3min* for ngram-count in the SRILM toolkit [Stolcke, 2002]) were set to 1 in all the LMs used in the following experiments.

4.5.1 Perplexity Experiments

We took the LM data sets from No.1 to No.5 in Table 4.1 as a core training set, named MTRAIN, which consists of 205,814 sentences and 1,847,201 words. We trained trigram IKN, MKN, HD, and HPY LMs using this training data. For the HDLM and the HPYLM, we ran 200 iterations sequentially on a single machine for inference, and collected 100 samples from the posterior over seating arrangements and hyperparameters.

The test data for perplexity estimation was extracted from the reference transcriptions for *rt06seval*. The final test data consisted of 3,597 sentences and 31,810 words.

[3] `http://corpus.amiproject.org`

Four different experimental conditions were considered and are shown in Table 4.3: the combination of whether or not a closed vocabulary (CV/OV) was used and/or mapping unknown words to a special symbol 'UNK' (+U) during training and testing. If the 'UNK' symbol is not included in the vocabulary, all unknown words will be skipped from the training data for the estimation and from the testing data for the evaluation.

Table 4.3 shows the perplexity results. We can see that in all four experiment conditions, the HPYLM has a lower perplexity than both the IKNLM and the MKNLM, and the HDLM has the highest perplexity, which is as expected and consistent with the previous results in [Teh, 2006b]. We used the CV condition, i.e., with a closed vocabulary but without the 'UNK' symbol, when training an LM in all the rest of experiments.

Table 4.3: Perplexity results on *rt06seval* testing data.

Condition	**VOCAB**	**UNK**	**IKNLM**	**MKNLM**	**HDLM**	**HPYLM**
OV	open	no	95.7	93.5	100.1	**88.6**
OV+U	open	yes	122.0	119.2	139.9	**111.9**
CV	close	no	110.1	106.5	118.5	**101.2**
CV+U	close	yes	110.5	106.8	120.4	**102.6**

Table 4.4 shows the perplexity results on the 5-fold cross validation of the scenario AMI Meeting Corpus, using the CV condition. The HPYLM again has a consistently lower perplexity than both the IKNLM and the MKNLM, and the HDLM again is the worst.

Table 4.4: Perplexity results of the 5-fold cross validation on the AMI scenario meetings.

FOLD	**IKNLM**	**MKNLM**	**HDLM**	**HPYLM**
1	107.1	104.5	116.6	**100.3**
2	99.3	97.1	107.8	**93.7**
3	106.5	103.9	115.8	**100.1**
4	101.8	99.5	110.3	**96.1**
5	101.1	98.5	108.2	**94.5**
average	103.2	100.7	111.7	**96.9**

4.5.2 ASR Experiments

We used the AMI-ASR system [Hain et al., 2007], described in Section 3.3, as the baseline platform for our ASR experiments. We only tested LMs trained using training data MTRAIN (see Table 4.1) under condition CV in Table 4.3, that is, we used a 50k vocabulary but without mapping unknown words to 'UNK' during training. For the HDLM and the HPYLM, we output an ARPA format LM. Different LMs were then used in the first pass decoding using `HDecode`[4].

Table 4.5 shows the WER results for the RT06s task. Unsurprisingly, the HDLM produces the highest WER. The HPYLM, however, results in a lower WER than both the IKNLM and the MKNLM. These reductions by the HPYLM from the IKNLM and the MKNLM are both significant using a matched-pair significance test [Jurafsky and Martin, 2009], with $p < 0.0005$ and $p < 0.02$ respectively. This is an encouraging result, since it is the first time that the HPYLM has been tested using a state-of-the-art large vocabulary ASR system on standard evaluation data.

Table 4.5: WER (%) results on *rt06seval* testing data.

LMS	SUB	DEL	INS	WER
IKNLM	15.7	9.9	2.9	28.5
MKNLM	15.6	10.0	2.8	28.4
HDLM	16.4	10.4	3.0	29.8
HPYLM	**15.3**	**10.1**	**2.7**	**28.1**

Table 4.6 shows the WER results for the 5-fold cross validation experiments on the scenario AMI meetings. We observed statistically significant ($p < 0.001$) reductions in WER by the HPYLM, which are consistent among all the five folds and the scenario AMI meetings as a whole. Our experiments also show that the HDLM consistently gives highest WERs. We therefore stop presenting results for the HDLM in the rest of experiments. We used only the transcriptions of the scenario AMI meetings to train LMs.

[4] `http://htk.eng.cam.ac.uk/`

Table 4.6: WER (%) results of 5-fold cross validation on the scenario AMI meetings. All the reductions by the HPYLM with respect to the IKNLM and the MKNLM are statistically significant, with $p < 0.001$.

FOLD	LMS	SUB	DEL	INS	WER
1	IKNLM	22.1	11.3	5.6	39.0
	MKNLM	21.9	11.4	5.5	38.9
	HPYLM	**21.7**	**11.5**	**5.4**	**38.6**
2	IKNLM	21.0	11.2	5.3	37.5
	MKNLM	20.8	11.4	5.2	37.4
	HPYLM	**20.7**	**11.6**	**5.0**	**37.3**
3	IKNLM	22.3	11.3	5.3	38.9
	MKNLM	22.2	11.4	5.1	38.7
	HPYLM	**21.9**	**11.5**	**4.9**	**38.3**
4	IKNLM	20.9	11.7	5.7	38.4
	MKNLM	20.9	11.3	5.6	37.8
	HPYLM	**20.7**	**11.3**	**5.5**	**37.5**
5	IKNLM	24.5	12.6	6.4	43.6
	MKNLM	24.2	13.4	6.1	43.7
	HPYLM	**24.1**	**12.9**	**6.1**	**43.1**
all	IKNLM	22.1	11.6	5.7	39.3
	MKNLM	21.9	11.7	5.5	39.1
	HPYLM	**21.7**	**11.7**	**5.3**	**38.8**

4.5.3 Scalability

To investigate the scalability of the HPYLM, we gradually increased the size of training data for the HPYLM, as shown in Table 4.7. MTRAIN includes the training data sets No.1–5. MTRAIN+F consists of MTRAIN and the No.6 data set Fisher-p1. Fur-

ther adding the data set No.7 to MTRAIN+F we obtained ALL-WC. Finally, we put together all the data No.1–8 as shown in Table 4.1, named ALL.

For MTRAIN, MTRAIN+F, and ALL-WC, experiments were carried out on a machine with dual quad-core Intel Xeon 2.8GHz processors and 12GB of memory. Table 4.7 shows the computational time per iteration and memory requirements when we change the size of training data, or vary the size of the vocabulary. From the results in Table 4.7, we can see that the inference for each iteration has time complexity of $O(n)$ when the vocabulary size is constant, where n is the size of training data. The smaller the size of the vocabulary, the quicker each iteration and the lower the memory requirement. The observations confirm the necessity of proposing a parallel training algorithm for the HPYLM. For IKNLM and MKNLM trained on ALL-WC, the memory requirement is around 1GB.

Table 4.7: Comparison of computational time and memory requirement of the HPYLM on different training data sets. Data set numbers refer to Table 4.1.

Name	**Data**	**#Words**	**Vocab**	**Time/Iter**	**Memory**
MTRAIN	No.1–5	1,847,201	50k	∼10sec	∼150MB
MTRAIN+F	No.1–6	12,440,604	50k	∼120sec	∼600MB
ALL-WC	No.1–7	48,514,322	50k	∼600sec	∼2400MB
			18k	∼300sec	∼2000MB
			8k	∼200sec	∼1400MB
ALL	No.1–8	211,427,888	50k	parallel	parallel

For ALL, it would be extremely demanding to train an HPYLM on this data set using a single machine, due to the computational time and memory limitations. We instead used the parallel training algorithm described in Section 4.3. We divided the inference into 50 sub-tasks, that is, we ran each iteration of the inference on 50 CPUs in parallel. It turns out that it is feasible to train an HPYLM on such a large corpus of more than 200 million word tokens, using a vocabulary of 50k work types in this way. The inference took around one day to finish 100 iterations, although this was highly dependent on submission and queueing times of the compute cluster. For ALL, we evaluated two HPYLM models – one after 32 Gibbs sampling iterations (HPYLM-iter32), and the other after 100 iterations (HPYLM-iter100).

We again evaluated perplexity performance over these four data sets to investigate the scalability of perplexity experiments. The perplexity results in Table 4.8 indicate that the HPYLM scales well to larger training data. We obtained consistent reductions in perplexity over both the IKNLM and the MKNLM. This further strengthens the perplexity results of Section 4.5.1. For two HPYLM models trained on ALL, we did not observe a significant difference in perplexity between HPYLM-iter32 and HPYLM-iter100.

Table 4.8: Perplexity results on *rt06seval* using different scale sizes of training data.

Data	**IKNLM**	**MKNLM**	**HPYLM**	
MTRAIN	110.1	106.5	**101.2**	
MTRAIN+F	106.7	103.6	**97.6**	
ALL-WC	102.9	100.8	**95.1**	
ALL	107.0	105.2	**99.3**(iter32)	**98.9**(iter100)

Finally we trained three types of ARPA format trigram LMs — IKNLM, MKNLM, and HPYLM — on both ALL-WC (a corpus of around 50 million word tokens) and ALL (a corpus of around 210 million word tokens) training data sets. Table 4.9 shows the WER results of these three different LMs in the first decoding using `HDecode`. On ALL-WC, we see the HPYLM performs slightly better than the IKNLM and the MKNLM. Significance testing shows the reductions by the HPYLM are not significant. On ALL, however, we observed significant reductions in WER by using the HPYLM, with $p < 0.0002$ and $p < 0.004$ for reductions over the IKNLM and MKNLM respectively. Once again, there is no significant difference in WER between HPYLM-iter32 and HPYLM-iter100.

4.5.4 Data Combination vs. Model Interpolation

Given several text corpora, i.e., those seven shown in Table 4.11, there are two different ways to estimate a language model. *Data combination* simply concatenates those seven corpora and trains a single language model on the combined corpus, without considering the differences between the corpora. This is the way in which we trained most LMs for the above experiments. *Model interpolation*, on the contrary, estimates seven separate language models on the corpora respectively, and linearly interpolates

Table 4.9: WER (%) results on *rt06seval* using different scale sizes of training data.

DATA	LMS	SUB	DEL	INS	WER
ALL-WC	IKNLM	14.6	10.0	2.6	27.3
	MKNLM	14.6	9.9	2.5	27.0
	HPYLM	**14.4**	**10.0**	**2.6**	**26.9**
ALL	IKNLM	14.5	9.7	2.7	27.0
	MKNLM	14.4	9.8	2.7	26.8
	HPYLM-iter32	**14.2**	**9.8**	**2.5**	**26.6**
	HPYLM-iter100	**14.2**	**9.8**	**2.6**	**26.5**

these seven LMs using some development data to optimize the interpolation weights. We have demonstrated the effectiveness of the HPYLM in the data combination style. Since model interpolation is commonly used in most state-of-the-art ASR systems, it is worthwhile for us to investigate the case of model interpolation.

We first investigated the experiments on *rt06seval*. We trained four sets of language models separately on MTRAIN, FISHER, WEBMEET, and WEBCONV in Table 4.1, with each set using interpolated Kneser-Ney, modified Kneser-Ney, and hierarchical Pitman-Yor process smoothing respectively. For each type of smoothing method, we interpolated the four separate language models using a development set of the evaluation data from NIST RT05s *rt05seval* (with 2,216 sentences, 16,282 word tokens). The final language models were obtained by interpolating with the optimal weights. Table 4.10 shows the perplexity and WER results for model interpolation on *rt06seval*. In comparison to data combination results for ALL (perplexity in Table 4.8 and WER in Table 4.9), we find model interpolation has both lower perplexity and lower WER. Considering the results for model interpolation, the HPYLM has slightly lower perplexity, but statistically insignificant WER, than the IKNLM and the MKNLM.

We additionally carried out the experiments on the 5-fold scenario AMI meetings in Table 4.2 to compare the two cases, using fold 2–5 as the training data and fold 1 as the testing data. We furthermore included six extra corpora in Table 4.11 as the training data for LMs. For data combination, we reached a combined corpus of 157 million word tokens in total. The parallel training algorithm was used to infer an HPYLM on this combined corpus, with 100 Gibbs sampling iterations. For model interpolation,

Table 4.10: Perplexity (PPL) and WER (%) results of comparing data combination and model interpolation on *rt06seval* testing data.

Experiment	LMS	PPL	SUB	DEL	INS	WER
Model	IKNLM	83.3	13.5	9.8	2.4	25.7
Interpolation	MKNLM	83.3	13.5	9.8	2.4	25.8
for *rt06seval*	HPYLM	**81.1**	**13.7**	**9.8**	**2.4**	**25.9**

we trained IKNLMs, MKNLMs, and HPYLMs (with 100 iterations) separately, using the seven training corpora respectively. Table 4.11 shows the perplexity results, indicating that the HPYLM consistently has a lower perplexity than the IKNLM and the MKNLM on each LM component. We used a 4-fold cross validation on folds 2–5 of the scenario AMI meetings to tune the optimal interpolated weights. Each time we took one fold from 2–5 as the development set, and the remaining three as the training data on which we trained the IKNLM, MKNLM and HPYLM respectively. The weights for the seven LM components were then optimized using the development set, for the IKNLM, MKNLM, and HPYLM respectively. We considered the average of the accumulated weights as the final optimal weights, which were used to estimate an interpolated LM.

Table 4.11: Statistics of corpora, and the perplexity results on the fold 1 of the scenario AMI meetings.

Data Sets	#Word Tokens	Perplexity		
		IKNLM	MKNLM	HPYLM
AMI fold2-5	771,870	109.3	106.4	**102.3**
ICSI	650,171	258.7	243.0	**234.0**
NIST-a	119,989	299.7	283.9	**277.5**
ISL-MCL	119,093	327.5	308.5	**304.0**
h5etrain03v1	3,494,444	234.6	223.3	**210.6**
Fisher-03-p1+p2	21,235,716	221.2	210.9	**200.7**
Hub4-lm96	130,896,536	321.1	301.3	**289.1**

Table 4.12 shows the perplexity and WER results on fold 1. It is not surprising to find that model interpolation is superior to data combination, because model interpolation weights the LM components of different domains to better match the testing data. Model interpolation provides significantly better results than data combination in perplexity and WER. We observed much higher perplexity results from data combination compared to model interpolation, due to the fact that a large portion of out-of-domain data (Hub4-lm96) was weighted identically to the in-domain meeting data in data combination. In either data combination or model interpolation, however, the HPYLM consistently has a lower perplexity result, and significantly ($p < 0.05$) lower WERs than the IKNLM and the MKNLM (although the absolute reductions are small for model interpolation). This suggests that we can train separate HPYLMs on several different corpora, and then use the standard method to interpolate these separate HPYLMs. This further consolidates our claim that the HPYLM is a better smoothing method than the IKNLM and the MKNLM for practical ASR tasks. It is more desirable, however, for a method to automatically weight and interpolate several HPYLMs directly within the hierarchical Pitman-Yor process framework. Wood and Teh [Wood and Teh, 2009] have proposed a model within the hierarchical Pitman-Yor process framework for domain adaptation. This approach, however, can only deal with two components, the in-domain LM and the out-of-domain LM, respectively. Additionally the computational complexity should be considered when doing interpolation for large corpora.

Table 4.12: Perplexity and WER (%) results of comparing data combination and model interpolation using the fold 1 of the scenario AMI meetings in Table 4.2 as the testing data.

Experiment	**LMS**	**PPL**	**SUB**	**DEL**	**INS**	**WER**
Data	IKNLM	168.6	22.2	10.7	5.7	38.6
Combination	MKNLM	163.9	22.0	10.8	5.6	38.5
for fold 1	HPYLM	**158.8**	**21.9**	**10.8**	**5.5**	**38.2**
Model	IKNLM	93.8	20.4	11.0	5.3	36.8
Interpolation	MKNLM	94.0	20.5	11.1	5.3	36.8
for fold 1	HPYLM	**91.4**	**20.4**	**11.2**	**5.1**	**36.7**

4.5.5 Experiments on Parallel training for HPYLM

In the following experiments, we trained HPYLMs using 100 iterations to burn in, and output standard ARPA format LMs, which were subsequently used in the first pass decoding using HDecode[5].

4.5.5.1 Meeting Corpora

The experiments reported in this section were performed using two meeting transcription tasks. The first task is the NIST Rich Transcription 2006 spring meeting recognition evaluations (RT06s). We tested only on those audio data recorded from individual headset microphones (IHM), consisting of meeting data collecting by the AMI project, CMU, NIST, and VT (Virginia Tech). The training data sets for language models used in this section, and the test transcription (*rt06seval*), are listed in Table 4.1. The webdata for meetings and conversational speech were collected from the world wide web using strategies described in [Wan and Hain, 2006].

Table 4.13: The statistics of the training and testing data sets for language models for the RT06s task.

No.	LM Data Set	#Sentences	#Words
1	*rt06strain*	205,814	1,847,201
2	Fisher (fisher-03-p1)	1,076,063	10,593,403
3	webdata (meetings)	3,218,066	36,073,718
4	webdata (conversational)	12,684,025	162,913,566
test	*rt06seval*	3,597	31,810

A second multiparty meeting corpus we consider in this section is a domain-specific meeting corpus — the AMI Meeting Corpus[6] [Carletta, 2007], as described in Section 3.1.1. We used the scenario part of the AMI Meeting Corpus for our experiments. There are 137 scenario meetings in total, of which we use 105 meetings (514,667 words) for training, and 32 meetings (175,302 words) for testing.

[5] http://htk.eng.cam.ac.uk/
[6] http://corpus.amiproject.org

4.5.5.2 Validation of Parallelization Errors

We first conducted experiments to validate the errors/losses caused by parallelization. We trained three HPYLMs of order 3 on fisher-03-p1 shown in Table 4.1, one without using parallel training algorithm but the traditional inference, and the other two with parallelism of IterSync and FinalSync, respectively. We output standard ARPA format LMs, and evaluated these LMs for the transcription of *rt06seval*. Table 4.14 shows perplexity and WER results, which claims that there is no statistically significant difference between these results, although FinalSync has a bit higher perplexity than the other two. It is therefore safe for us to use the proposed parallel training algorithm for the HPYLM. In the following experiments, we all use the IterSync parallel training algorithm for inferring HPYLMs.

Table 4.14: Results of training on fisher-03-p1 and testing on *rt06seval*, to compare three cases of training an HPYLM: without parallelization, IterSync parallelism, and FinalSync parallelism, respectively.

Parallel	PPL	SUB	DEL	INS	WER
No	121.4	16.0	9.8	3.0	28.9
IterSync	121.8	16.1	9.8	3.0	28.9
FinalSync	126.6	16.1	9.7	3.1	28.9

To measure experimentally the errors introduced by the parallel training algorithm, we trained two HPYLMs of order 3 on FISHER shown in Table 4.1 for 100 iterations, one without parallelism, while the other with the parallel training algorithm introduced in Section 4.3. We evaluated the two HPYLMs for the transcription of *rt06seval*, recording the perplexity results for each iteration. Figure 4.3 shows that there is no statistically significant difference between perplexity results of these two HPYLMs, which indicates that the error caused by the approximation in parallel training algorithm can be ignored. We also find no significant difference in the WER [Huang and Renals, 2009].

In summary, the parallel training algorithm largely decreases the time for inferring an HPYLM comparing to a sequential training algorithm, and there is no significant difference in model performance in terms of perplexity and WER. However, there are still some open issues with the parallel training algorithm for the HPYLM, for example, different runs of the parallel training algorithm may result in slightly different results,

and there is an issue due to imbalanced jobs submitted — one delayed job may slow down the whole inference.

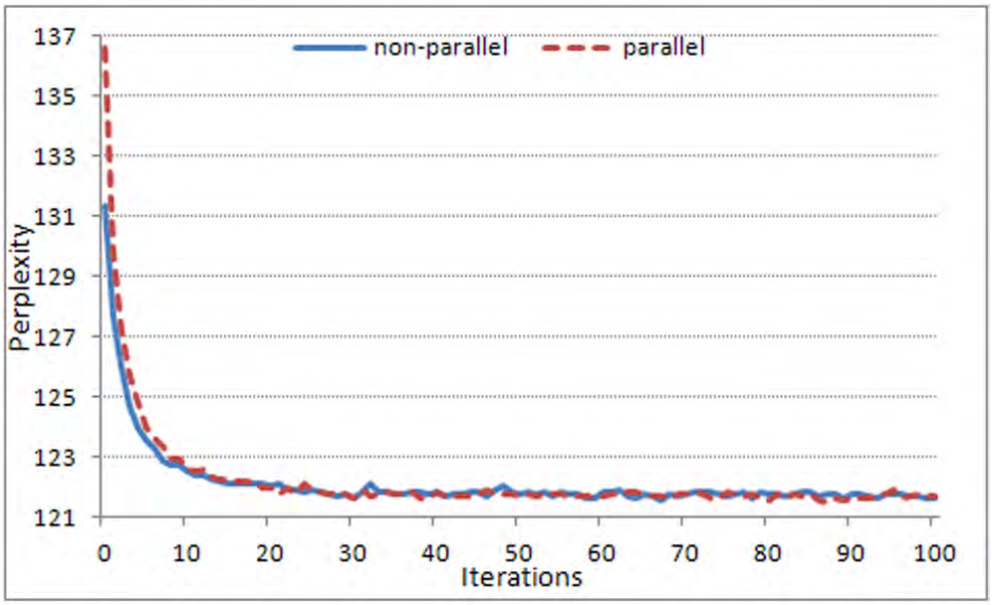


Figure 4.3: Perplexity results by iterations, trained on FISHER and tested on RT06s evaluation data *rt06seval* for the HPYLM, to validate parallelization approximation.

4.5.5.3 Experiments on *rt06seval*

We trained three types of ARPA format trigram LMs – IKNLM, MKNLM, and HPYLM – on data sets No. 1–4 in Table 4.1, which is overall a corpus of around 210 million words. We observe in Table 4.15 that the HPYLM has a lower perplexity than both IKNLM and MKNLM. Moreover, the HPYLM produces significant lower WER ($p < 0.005$) compared with both the IKNLM and the MKNLM.

Table 4.15: The perplexity and word error rate results on *rt06seval*.

LMS	PPL	SUB	DEL	INS	WER
IKNLM	107.0	14.5	9.7	2.7	27.0
MKNLM	105.2	14.4	9.8	2.7	26.8
HPYLM	**98.9**	**14.2**	**9.8**	**2.6**	**26.5**

4.5.5.4 Experiments on AMI Corpus

In addition to the training data from the scenario AMI meetings , we included broadcast news (HUB-4) and conversational text (Fisher), which totally makes up a corpus of around 200 millions of words for training LMs. Again we trained IKNLM, MKNLM, and HPYLM on this combined data, and used these three LMs for the transcription of the testing data of 32 scenario AMI meetings . It is not surprising to find that the HPYLM is more accurate than both the IKNLM and the MKNLM, with lower perplexity and significantly lower WER ($p < 0.005$).

Table 4.16: The perplexity and word error rate results on the scenario AMI Meeting Corpus.

LMS	PPL	SUB	DEL	INS	WER
IKNLM	168.6	22.2	10.7	5.7	38.6
MKNLM	163.9	22.0	10.8	5.6	38.5
HPYLM	**158.8**	**21.9**	**10.8**	**5.5**	**38.2**

4.6 Analysis and Discussion

4.6.1 Convergence

It is often expensive to train an HPYLM, especially when working with large training corpora as demonstrated in Table 4.7. Therefore the convergence of HPYLM is an important factor. We trained an HPYLM using the data set MTRAIN in Table 4.1. During each iteration, we collected the log likelihood over the training data, and the predictive log likelihood over the testing data *rt06seval*. Figure 4.4 shows the convergence of likelihoods over training and testing data for the first 150 iterations. From this we can see that after about twenty iterations, the HPYLM has quickly converged to a lower predictive log likelihood value on the testing data, which roughly remains the same for further iterations. On the other hand, although it is slow to train an HPYLM on large corpora, we only need to train the model once and output an ARPA format LM, then apply it in an ASR system as a standard n-gram LM. We also observed that the likelihood over the training data decreases after more and more iterations, while

the likelihood over the testing data increases, which means the generalisation of the HPYLM improves.

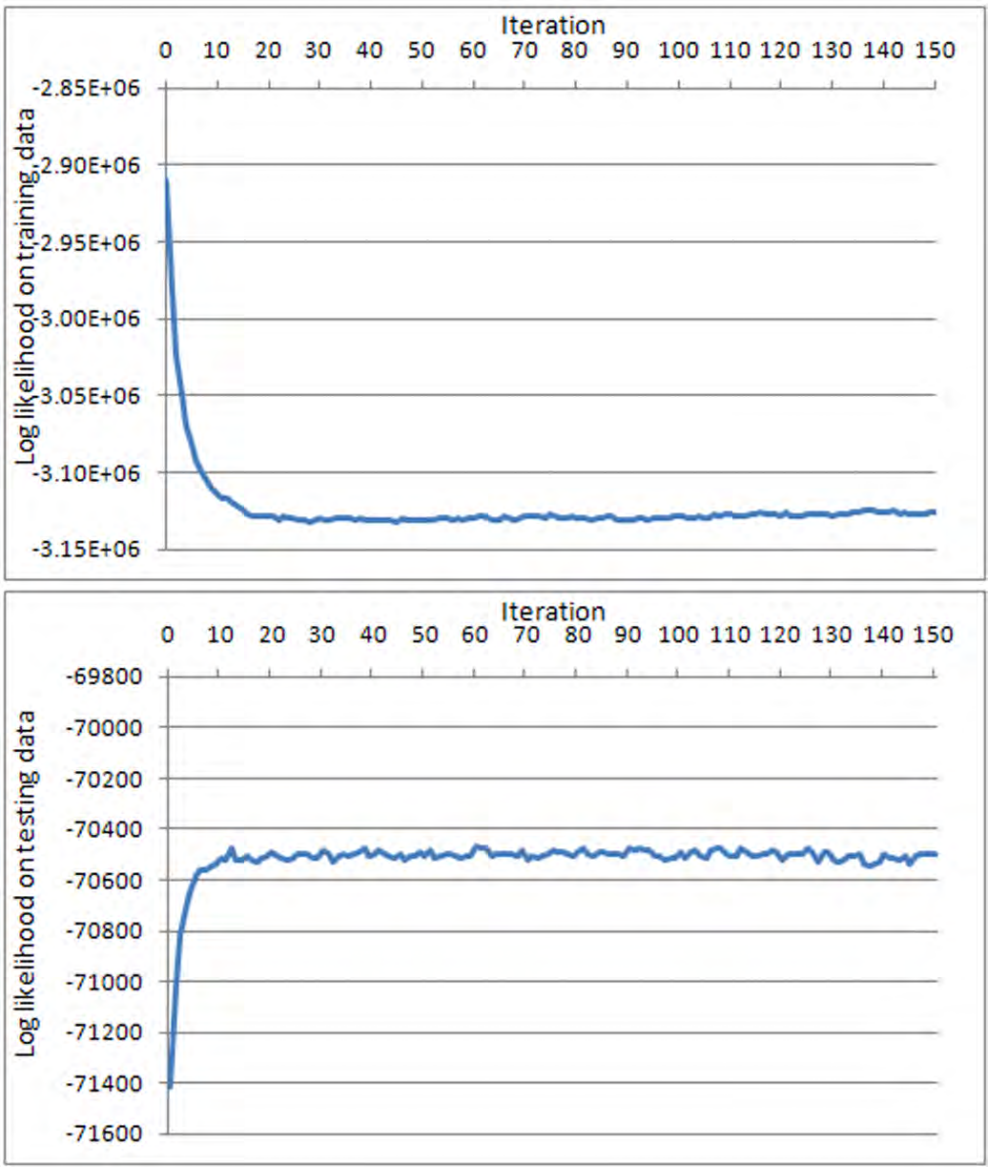


Figure 4.4: Convergence of the HPYLM. The log likelihood over the training data MTRAIN (top), and the log likelihood over the testing data *rt06seval* (bottom), to investigate the convergence of the HPYLM with iterations.

The finding is further confirmed by from the experimental perplexity and WER results in Table 4.8 and Table 4.9 respectively for two HPYLM models, one from the 32nd iteration and the other from the 100th iteration. There is no significant difference between these two models, which indicates normally we need only tens of iterations to infer an HPYLM model, since the posterior of the HPYLM is well-behaved.

4.6.2 Average Discounts

For a specific order of an n-gram LM, there is only one discount $d_{|\mathbf{u}|}$ for the IKNLM. The MKNLM, instead, has three different discount parameters, $d_{|\mathbf{u}|1}$, $d_{|\mathbf{u}|2}$, and $d_{|\mathbf{u}|3+}$, according to the counts of n-grams. This additional flexibility of discounting in the MKNLM has been proven to be superior in practice to the IKNLM [Chen and Goodman, 1999]. We find from Equation (4.4) that the HPYLM has a more relaxed discounting scheme, by allowing each context to use a different discounting value: $d_{|\mathbf{u}|}t_{\mathbf{u}w}$. It was noted by Teh [Teh, 2006a] that the expected number of tables $t_{\mathbf{u}w}$ in a Pitman-Yor process scales as $O(c_{\mathbf{u}w}^{d_{|\mathbf{u}|}})$ where $c_{\mathbf{u}w}$ is the number of customers and $d_{|\mathbf{u}|}$ is the discount parameter of the Pitman-Yor process. Therefore, the actual amount of discount, $d_{|\mathbf{u}|}t_{\mathbf{u}w}$, will grow slowly as the count $c_{\mathbf{u}w}$ grows.

To demonstrate the rich-get-richer property of discounting in the HPYLM, we investigate an HPYLM trained for 300 iterations using MTRAIN (meeting data) in Table 4.7, and plot average discounts as a function of trigram counts in Figure 4.5. We additionally plot on the bottom of Figure 4.5 the *counts of counts* statistics used to do the averaging over discounts. The counts of counts histogram exhibits a long-tailed distribution: most words occur with lower counts, while only a few words occur frequently. We find that the growth of average discounts in the HPYLM is sublinear, with respect to counts of trigrams. For a comparison, the IKNLM uses a single discounting parameter: $d_3 = 0.8083$, and the MKNLM uses three different discounts: $d_{31} = 0.8083, d_{32} = 1.1428, d_{33+} = 1.3551$. It is interesting to note that discounts in the HPYLM for trigrams with large values of counts are also surprisingly large, which reminds us to question whether or not it is enough to use only one discount parameter $d_{|\mathbf{u}|3+}$ in the MKNLM for all n-grams with three or more counts. Finally we note that similar counts may have substantial differences in average discounts, especially in the case of larger counts. This may arise from local effects during the sample-based inference of the seating arrangements—more likely in a restaurant with more customers (larger counts)—-and also because discounts for larger counts are averaged over fewer trigrams (count of counts changes inversely with counts), which makes it look less smoothed for large counts.

4.6.3 Learning Curve

To further verify that the HPYLM is a suitable choice for language modelling when there is a large amount of data, we investigated the learning curve of the HPYLM with

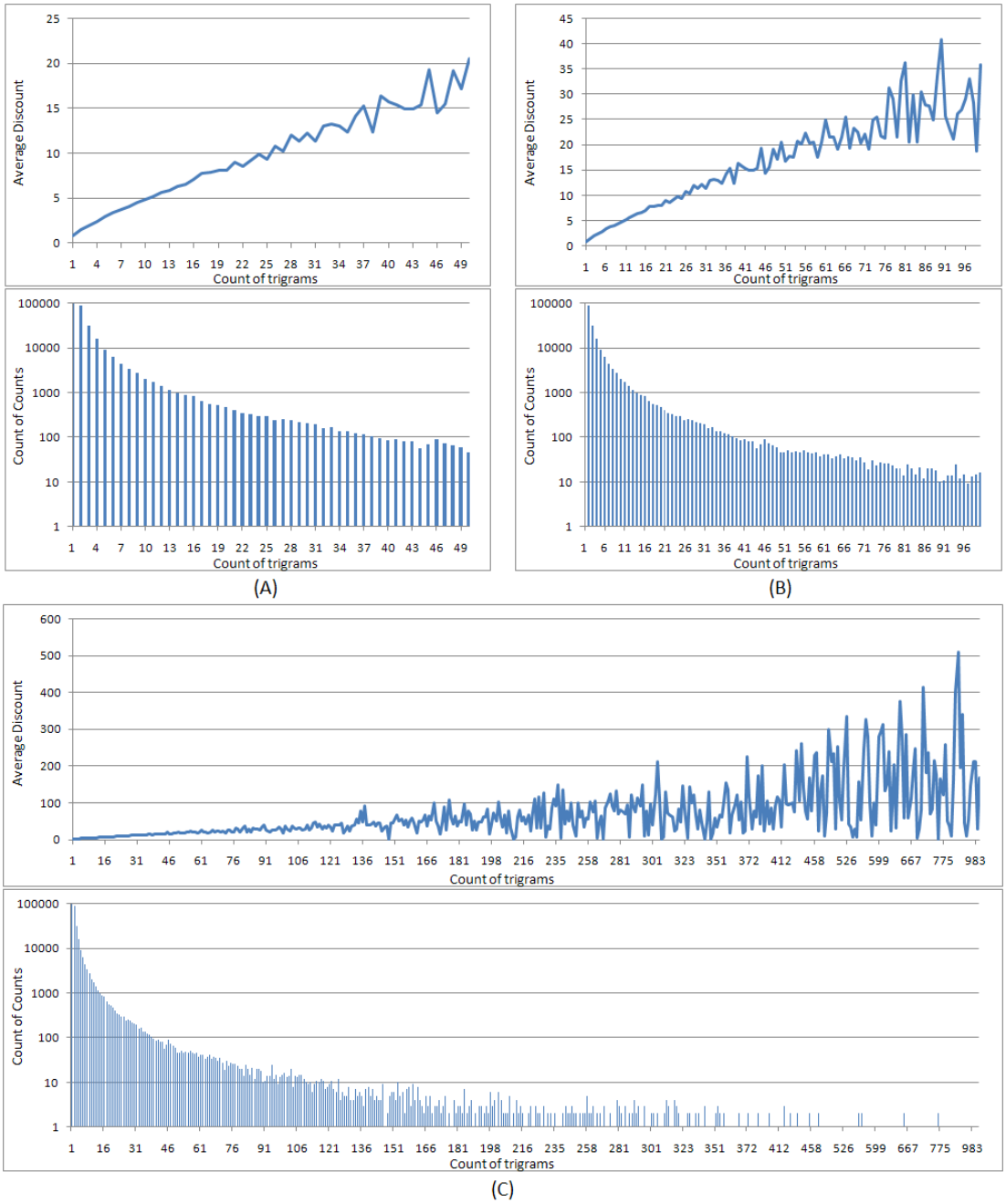


Figure 4.5: Average discounts as a function of trigram counts (top), and counts of trigram counts (bottom) in the HPYLM trained on the data set MTRAIN, with different scales for horizontal axis: first 50 counts (A), first 100 counts (B), and first 1000 counts(C).

respect to the amount of training data. We considered ALL, a corpus of 211 million word tokens consisting of data sets No.1–8 in Table 4.1. We first randomly reordered the corpus, and then varied the training set size between 1 million and 211 million words by gradual increments to obtained training sets of 1, 10, 25, 50, 100, 200, and 211 million words. We trained the IKNLM, MKNLM, and HPYLM using the CV condition on these training sets respectively, and evaluated the language models on the RT06s test data (*rt06seval*), as in Section 4.5.1. Figure 4.6 shows the learning curve of perplexity results as the amount of training data increases. We see that, although there is a runtime overhead, the HPYLM consistently outperforms the IKNLM and the MKNLM as the size of training data increases. Moreover, we see in Figure 4.6 that the reductions by the HPYLM become larger with larger amount of training data available. Generally, the larger the training data available, the smaller the effect of smoothing techniques for language modelling. The above observation is partly due to the data partition we used in the experiment.

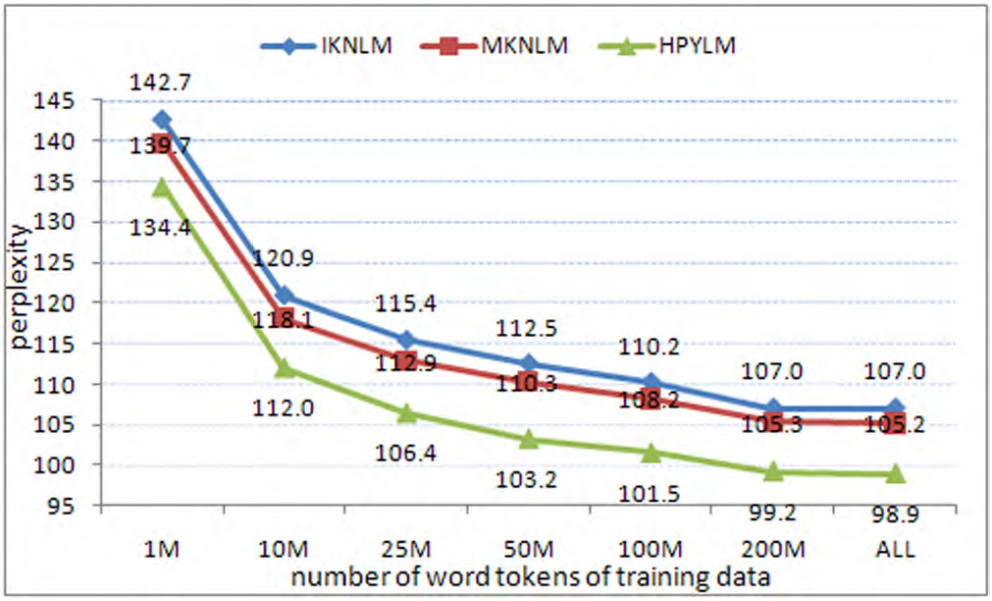


Figure 4.6: Learning curve of perplexity results on RT06s evaluation data *rt06seval* for the HPYLM with respect to the amount of training data.

4.6.4 Comparison to Other Models

In our experiments, the HPYLM produces consistently better perplexity results than the MKNLM. This observation is in contrast to the finding in [Teh, 2006b], where the HPYLM performs slightly worse than the MKNLM in terms of perplexity. We

argue that there are two potential reasons for this. Firstly, Teh [2006b] used conjugate gradient descent in the cross-entropy on the validation set to determine the discounting parameters for the IKNLM and the MKNLM, which is a different approach to that used in most standard language model toolkits such as SRILM [Stolcke, 2002]. We used in this thesis the SRILM toolkit to build reasonable baseline results and evaluate our various LMs. Secondly, the difference in the data and the implementation may be another reason. Based on the discussion in Section 4.6.2, however, we believe that it makes sense for the HPYLM to outperform the MKNLM, because of its more flexible discounting scheme for each different context.

There are different interpretations for the unusual modified counts for lower order n-grams in the IKNLM. Kneser and Ney [Kneser and Ney, 1995], and Chen and Goodman [Chen and Goodman, 1999], derived the modified counts for lower order n-grams in terms of preserving marginal word distribution constraints. Goodman [Goodman, 2004] justified this from a Bayesian view of a maximum entropy model with an exponential prior, and clearly explained why the Kneser-Ney smoothing has that particular form including the modified counts for lower order n-grams. The maximum entropy model, to which the back-off version of Kneser-Ney smoothing forms an approximation, also preserves the marginal constraints. The HPYLM is a full Bayesian interpretation of interpolated Kneser-Ney as approximate inference in a hierarchical Bayesian model consisting of Pitman-Yor processes [Teh, 2006b]. The modified counts for lower order n-grams, $c_{\mathbf{u}w\bullet}$, are derived directly from Chinese restaurant processes and different from those in the IKNLM, subject to the following relationships among $c_{\mathbf{u}w\bullet}$ and $t_{\mathbf{u}w\bullet}$ [Teh, 2006b]:

$$\begin{cases} t_{\mathbf{u}w\bullet} = 0 & \text{if } c_{\mathbf{u}w\bullet} = 0; \\ 1 \leq t_{\mathbf{u}w\bullet} \leq c_{\mathbf{u}w\bullet} & \text{if } c_{\mathbf{u}w\bullet} > 0; \end{cases} \qquad c_{\mathbf{u}w\bullet} = \sum_{\mathbf{u}':\pi(\mathbf{u}')=\mathbf{u}} t_{\mathbf{u}'w\bullet}$$

where $t_{\mathbf{u}'w\bullet}$ is the number of tables seated by $c_{\mathbf{u}'w\bullet}$ customers in the child restaurant of $\mathbf{u}$. The value of $t_{\mathbf{u}'w\bullet}$, and consequently the modified count $c_{\mathbf{u}w\bullet}$, are naturally determined by the seating arrangements induced from Chinese restaurant processes. If we constrain $t_{\mathbf{u}'w\bullet} = 1$ if $c_{\mathbf{u}w\bullet} > 0$, the modified count $c_{\mathbf{u}w\bullet}$ is the same as that in the IKNLM. However, the HPYLM also satisfies marginal constraints when the strength parameter $\theta_m = 0$ for all $m < n$, as proved in [Teh, 2006a].

The structural Bayesian language model by Yaman *et al.* [Yaman et al., 2007] shares similar ideas to the hierarchical Dirichlet language model [MacKay and Peto, 1994] (although it is estimated by a MAP approach), in that both models assemble n-grams in a tree structure and use the Dirichlet distribution as the prior to smooth the

empirical counts. In comparison to the HPYLM, however, both models suffer from performance improvements compared to state-of-the-art smoothing methods such as IKN and MKN, for lack of one important issue for language model smoothing - absolute discounting.

4.7 Summary

In this chapter, we present the application of hierarchical Pitman-Yor process language models on a large vocabulary ASR system for conversational speech, using reasonably large corpora. We show comprehensive experimental results on multiparty conversational meeting corpora, with the observation that the HPYLM outperforms both the IKNLM and the MKNLM.

Overall, we hope to convey our judgment that it is feasible, and worthwhile, to use the HPYLM for applications in large vocabulary ASR systems. In detail, the conclusions we make in this chapter are as follows: 1) the HPYLM provides an alternative interpretation, Bayesian inference, to language modelling, which can be reduced to interpolated Kneser-Ney smoothing [Teh, 2006b]; 2) the HPYLM provides a better smoothing algorithm for language modelling in practice, which has better perplexity and WER results than both the IKNLM and the MKNLM, consistent and significant; 3) HPYLM training converges in relatively quickly; 4) a parallel training scheme makes it possible to estimate models in the case of large training corpora and large vocabularies.

The main contributions of the HPYLM work in this chapter include the introduction of a novel Bayesian language modelling technique to the ASR community, and the experimental verification on the task of large vocabulary ASR for conversational speech in meetings. We have demonstrated that it is feasible to infer a hierarchical non-parametric Bayesian language model from a large corpus, thus making it practical to use for large vocabulary speech recognition or machine translation. Our experimental results have shown that any approximations resulting from the parallel algorithm have a negligible effect on performance. Overall, the HPYLM results in significantly improved accuracy compared with the current state-of-the-art (IKNLM/MKNLM). The resulting language model may be interpreted as a smoothed n-gram model, can be implemented in a standard way (e.g., using an ARPA format language model file), and may be used in place of other smoothed n-gram language models.

In conclusion we have demonstrated that it is feasible to infer a hierarchical non-parametric Bayesian language model from a large corpus, thus making it practical to

use for large vocabulary speech recognition or machine translation. Our experimental results have shown that any approximations resulting from the parallel algorithm have a negligible effect on performance. Overall, the HPYLM results in significantly improved accuracy compared with the current state-of-the-art (IKNLM/MKNLM). The resulting language model may be interpreted as a smoothed n-gram model, can be implemented in a standard way (e.g., using an ARPA format language model file), and may be used in place of other smoothed n-gram language models.

Chapter 5

Power Law Discounting Language Models

In this chapter, we present an approximation to the Bayesian hierarchical Pitman-Yor process language model which maintains the power law distribution over word tokens, while not requiring a computationally expensive approximate inference process. This approximation, which we term power law discounting, has a similar computational complexity to interpolated and modified Kneser-Ney smoothing. We performed experiments on meeting transcription using the NIST RT06s evaluation data and the AMI corpus, with a vocabulary of 50,000 words and a language model training set of up to 211 million words. Our results indicate that power law discounting results in statistically significant reductions in perplexity and word error rate compared to both interpolated and modified Kneser-Ney smoothing, while producing similar results to the hierarchical Pitman-Yor process language model.

5.1 Introduction

One of the most remarkable statistical properties of word frequencies in natural language is the fact that they follow a power law distribution, i.e., $P(c_w = x) \propto x^{-d}$ where c_w is the number of occurrences of word w in a corpus and d is a constant. The well-known Zipf's law equivalently states the fact in terms of the frequency ranking of words: a few outcomes have very high probability while most outcomes occur with low probability. It follows that a stochastic process, such as the Pitman-Yor process [Pitman and Yor, 1997], that has the "rich-get-richer" capacity to generate a power law distribution is able to take advantage of this property for natural language modelling.

In Chapter 4, we introduced the HPYLM, which offers a hierarchical Bayesian approach to language modelling, with a non-parametric prior distribution. Since the HPYLM is a computationally expensive estimate, we present an approximation to it, that we call *power law discounting* in which the discounting parameters have a direct functional form and do not require approximate inference. Inference in power law discounting has a similar computational complexity to interpolated or modified Kneser-Ney. We evaluate this new approach to language model smoothing in terms of perplexity and WER in the domain of multiparty meetings, reporting results on the NIST RT06s evaluation data and on the AMI corpus. Our results indicate that power law discounting is a good approximation to the more computationally expensive HPYLM, and maintains statistically significant decreases in WER and perplexity compared with interpolated and modified Kneser-Ney smoothing. Finally we present some analysis of the power law discounting scheme, including a demonstration of the power law property, as well as an investigation of the effect of the discounting parameters.

5.2 Power Law Discounting

In this section we introduce an efficient approximation to the HPYLM, in which $t_{\mathbf{u}w}$ the number of tables occupied by word w in the restaurant corresponding to context $\mathbf{u}$ is approximated as a function of $c_{\mathbf{u}w}$, the count of occurrences of word w following context $\mathbf{u}$. This is in contrast to the HPYLM in which this parameter is obtained using an approximate inference scheme based on Markov Chain Monte Carlo sampling.

We simplify the notations for $d_{|\mathbf{u}|}$ and $\theta_{|\mathbf{u}|}$ by ignoring the subscript $|\mathbf{u}|$. In the HPYLM, the $c_{\mathbf{u}w}$ occurrences of a word w following context $\mathbf{u}$ is discounted by $dt_{\mathbf{u}w}$, i.e., $\hat{c}_{\mathbf{u}w} = c_{\mathbf{u}w} - dt_{\mathbf{u}w}$. The approximate inference of $t_{\mathbf{u}w}$ is time and memory intensive. However, the expected number of tables $\mathbb{E}(t_{\mathbf{u}\bullet})$ in a Pitman-Yor process used in the HPYLM follows a power law growth with $c_{\mathbf{u}\bullet}$ where $\bullet$ denotes the marginal operation [Teh, 2006a]. Based on this observation, we therefore propose a power law

discounting LM (PLDLM) which smoothes n-grams as follows:

$$d = \frac{n_1}{n_1 + 2n_2} \tag{5.1}$$

$$t_{\mathbf{u}w} = f(c_{\mathbf{u}w}) = c_{\mathbf{u}w}^d \tag{5.2}$$

$$t_{\mathbf{u}\bullet} = \sum_w t_{\mathbf{u}w} = \sum_w c_{\mathbf{u}w}^d \tag{5.3}$$

$$P^{\text{PLD}}(w|\mathbf{u}) = \frac{\max(c_{\mathbf{u}w} - dt_{\mathbf{u}w}, 0)}{\theta + c_{\mathbf{u}\bullet}} + \frac{\theta + dt_{\mathbf{u}\bullet}}{\theta + c_{\mathbf{u}\bullet}} P^{\text{PLD}}(w|\pi(\mathbf{u})) \tag{5.4}$$

The parameter d corresponds to the traditional discount parameter in the IKNLM for $(|\mathbf{u}|+1)$-grams, n_1 and n_2 are the total number of n-grams with exactly one and two counts and θ is the strength parameter. The estimate of $t_{\mathbf{u}w}$ in Equation (5.2) significantly simplifies the model while maintaining the most important part of the HPYLM: the power law characteristic. Another key issue for the PLDLM is the modified counts $c_{\mathbf{u}'w}$ for lower order $(n-1)$-grams of context $\mathbf{u}'$, as shown in Equation (5.6).

$$\begin{cases} t_{\mathbf{u}w} = 0 & \text{if } c_{\mathbf{u}w} = 0; \\ 1 \leq t_{\mathbf{u}w} < c_{\mathbf{u}w} & \text{if } c_{\mathbf{u}w} > 0; \end{cases} \tag{5.5}$$

$$c_{\mathbf{u}'w} = \sum_{\mathbf{u}:\pi(\mathbf{u})=\mathbf{u}'} t_{\mathbf{u}w} = \sum_{w'} t_{w'\mathbf{u}'w} \tag{5.6}$$

The strength parameter θ can be estimated using the same technique as that for the HPYLM, i.e., a sampling method based on auxiliary variables [Teh, 2006a]. Since the model is insensitive to θ, as demonstrated in Section 5.5.2, we can alternatively set the values empirically, or simply ignore θ in Equation (5.4). In our experiments in section 5.4 we set $\theta = 0$.

We can decide on the number of discount parameters, denoted as p, to be used in the PLDLM, i.e., using one for each for $c_{\mathbf{u}w} = 1, 2, \ldots, p-1$ and another for all $c_{\mathbf{u}w} \geq p$. The PLDLM exactly reduces to the IKNLM when $p = 1$. The PLDLM with $p = 3$, PLD_3, is directly comparable to the MKNLM since both have three free discount parameters, except that the PLDLM takes a more straightforward form ($dc_{\mathbf{u}w}^d$) than the MKNLM [Chen and Goodman, 1999]. The amount of discount in the PLDLM is a function of counts $c_{\mathbf{u}w}$, which grows slowly as the count increases.

5.3 Marginal Constraints

Kneser and Ney [Kneser and Ney, 1995] demonstrated the importance of preserving the marginal constraints in language modelling. In Section 2.2.1.4, we noted that the

Kneser-Ney smoothing [Kneser and Ney, 1995] was motivated to follow the marginal constraints by using a modified distribution for lower order n-grams. Following [Chen and Goodman, 1999; Teh, 2006a], we show that the PLDLM satisfies the marginal constraints when the strength parameter $\theta = 0$, and we use the predictive probability in Equation (5.4) and modified counts in Equation (5.6):

$$\frac{c_{\mathbf{u}'w}}{c_{\mathbf{u}'\bullet}} = \sum_{w'} \frac{c_{w'\mathbf{u}'}}{c_{\bullet\mathbf{u}'}} P^{\text{PLD}}_{w'\mathbf{u}'}(w) \tag{5.7}$$

$$\begin{aligned} c_{\mathbf{u}'w} &= \sum_{w'} c_{w'\mathbf{u}'} \left[\frac{c_{w'\mathbf{u}'w} - dt_{w'\mathbf{u}'w}}{c_{w'\mathbf{u}'}} + \frac{dt_{w'\mathbf{u}'} P^{\text{PLD}}_{\mathbf{u}'}(w)}{c_{w'\mathbf{u}'}} \right] \\ &= \sum_{w'} \left(c_{w'\mathbf{u}'w} - dt_{w'\mathbf{u}'w} + dt_{w'\mathbf{u}'} P^{\text{PLD}}_{\mathbf{u}'}(w) \right) \\ &= c_{\mathbf{u}'w} - dt_{\mathbf{u}'w} + dt_{\mathbf{u}'} P^{\text{PLD}}_{\mathbf{u}'}(w) \end{aligned} \tag{5.8}$$

If we solve this and apply Equation (5.6) we obtain:

$$P^{\text{PLD}}_{\mathbf{u}'}(w) = \frac{t_{\mathbf{u}'w}}{t_{\mathbf{u}'}} = \frac{\sum_{w'} t_{w'\mathbf{u}'w}}{\sum_{w} \sum_{w'} t_{w'\mathbf{u}'w}} = \frac{c_{\mathbf{u}'w}}{c_{\mathbf{u}'}} \tag{5.9}$$

5.4 Experiments and Results

We have evaluated the PLDLM using two meeting transcription tasks: the NIST Rich Transcription 2006 spring meeting evaluation (RT06s), and the scenario portion of the AMI meeting corpus. In each case we compared the PLDLM to the IKNLM, the MKNLM and the HPYLM. We trained the all following trigram LMs using cutoff values for counts of 1, by using the SRILM toolkit [Stolcke, 2002] and the PLDLM program [1]. We did not use the strength parameter θ when training PLDLMs, i.e., we set $\theta = 0$ in Equation (5.4). We used the AMI-ASR system [Hain et al., 2007] as the baseline platform for our ASR experiments, using all LMs in the first pass decoding.

5.4.1 NIST Rich Transcription 2006 Evaluation

For the RT06s task we trained LMs on 1.8M words of transcribed meetings data (meetings-s1), 10.6M words of transcribed conversationaltelephone speech (fisher-03-p1), and web data matched to meeting (webmeet; 36.1M words) and conversational (webconv; 162.9M words) speech collected using the approach described by Wan and Hain [2006]. In total this resulted in 211.4M words of LM training data (ALL-1). We

[1] The toolkit for power law discounting language model is available from `http://homepages.inf.ed.ac.uk/s0562315/`

performed experiments using a vocabulary of 50,000 words. Table 5.1 shows the perplexity results on the NIST RT06s test data *rt06seval* (31,810 words). We found that, in all cases, the PLDLM outperforms the IKNLM and the MKNLM, and has comparably similar results to the HPYLM. The PLDLM with three discount parameters (PLD_3) also results in a slightly lower perplexity compared to the MKNLM.

Table 5.1: Perplexity results on NIST RT06s *rt06seval*.

DATA	IKN	MKN	HPY	PLD_3	PLD
meeting-s1	110.1	106.5	101.2	105.7	**104.3**
fisher-03-p1	134.0	128.5	121.4	128.1	**122.6**
webmeet	176.8	170.6	159.3	169.6	**159.7**
webconv	135.4	131.8	120.2	130.5	**120.8**
ALL-1	107.0	105.2	98.9	104.6	**100.7**

Table 5.2 shows the speech recognition WERs for *rt06seval* using LMs trained on ALL-1. The PLDLM is significantly better than the IKNLM and the MKNLM (weak), with $p < 0.01$ and $p < 0.05$ respectively, but is not significantly different to the HPYLM. We also evaluated using individual data sets, and found that there is no significant reduction in WER on meeting-s1 and fisher-03-p1, but there are significant reductions in WER by the PLDLMs on the webmeetings and webconv conditions, compared to the MKNLM. This may suggest that the increment of discounts in the PLDLM in turn increases the generalisation ability of LMs in if the domain is somewhat mismatched.

Table 5.2: WER (%) results on NIST RT06s *rt06seval*.

LMS	SUB	DEL	INS	WER
IKNLM	14.5	9.7	2.7	27.0
MKNLM	14.4	9.8	2.7	26.8
HPYLM	14.2	9.8	2.6	26.5
PLDLM	**14.2**	**10.0**	**2.4**	**26.6**

5.4.2 The AMI Corpus

For the AMI corpus, we trained LMs on 1.7M words of meeting transcripts (meeting-s2), 3.5M words of conversational speech transcripts (h5etrain03v1), a further 21.2M words of conversational speech transcripts (fisher-03-p1+p2), and 130.9M words of broadcast news transcripts (hub4-lm96), totalling 157.3M words of LM training data (ALL-2). Again we used a vocabulary of 50,000 words in our experiments. Table 5.3 shows the perplexity results on a test set of 32 AMI scenario meetings *amieval* (175,302 words). We found similar observations as those for *rt06seval*. Moreover, the PLDLM even slightly outperforms the HPYLM on some corpora, i.e., on fisher-03-p1+p2 and hub4-lm96.

Table 5.3: Perplexity results on the AMI meetings *amieval*.

DATA	IKN	MKN	HPY	PLD_3	PLD
meeting-s2	114.7	112.0	106.9	110.9	**110.9**
h5etrain03v1	234.6	223.3	210.6	220.5	**210.8**
fisher-03-p1+p2	221.2	210.9	200.7	209.7	**198.2**
hub4-lm96	321.1	301.3	289.1	303.3	**282.5**
ALL-2	168.6	163.9	158.8	163.7	**157.9**

Table 5.4 shows the speech recognition WERs on *amieval* using LMs trained on ALL-2. The reduction in WER by the PLDLM is significant comparing to the IKNLM and the MKNLM, with $p < 0.001$. Again there is no significant difference between the PLDLM and the HPYLM ($p < 0.15$), which to some extent implies that the PLDLM well approximates the HPYLM.

Table 5.4: WER (%) results on the AMI meetings *amieval*.

LMS	SUB	DEL	INS	WER
IKNLM	22.2	10.7	5.7	38.6
MKNLM	22.0	10.8	5.6	38.5
HPYLM	21.9	10.8	5.5	38.2
PLDLM	**22.0**	**10.9**	**5.5**	**38.3**

5.5 Analysis and Discussions

5.5.1 Absolute Discount

To demonstrate the power law property of discounts in the PLDLM and the HPYLM, we trained an PLDLM using meeting-s1, and an HPYLM for 300 iterations on the same data. We plotted average discounts as a function of trigram counts in Figure 5.1.

We have discussed the properties of absolute discounts in the HPYLM in Section 4.6.2. The average discounts of the PLDLM approaches to the expected values of discounts in the HPYLM with a smoothed curve. We note that trigrams with large values of counts in the PLDLM have large discounts, which is different from the intuition for the IKN/MKN smoothing, and could potentially overestimate the discounts in some case. However, this may not be a big problem, because those trigrams with large values of counts occur only infrequently in the training corpus.

5.5.2 Effect of Strength Parameter θ

To study the effect of strength parameter θ, we trained a trigram PLDLM using θ on fisher-03-p1, which has lower perplexity (121.7) than the PLDLM without θ (122.6). The initial values of θ for were obtained using the auxiliary sampling method, resulting in $\theta_1 = 3.3$, $\theta_2 = 2.25$, and $\theta_3 = 2.0$. Each time we ranged one θ over 0 to 100, while keeping the other two fixed. We evaluated the perplexity on *rt06seval* as the value of θ change. Results in Figure 5.2 show that the PLDLM is more sensitive to θ_2 and θ_3 than θ_1, and smaller values of θ_2 and θ_3 work better.

5.5.3 Effect of Discount Parameter Numbers

To investigate the effect of discount parameter numbers, we trained various trigram PLDLMs on fisher-03-p1+p2 by setting p from 1 to ∞ (none), and evaluated perplexity on *amieval*. As shown in Figure 5.3, the perplexity decreases as the number of free discount parameters increases, which implies that it is better to use more discount parameters if we have a coherent way to estimate them, as done in the PLDLM.

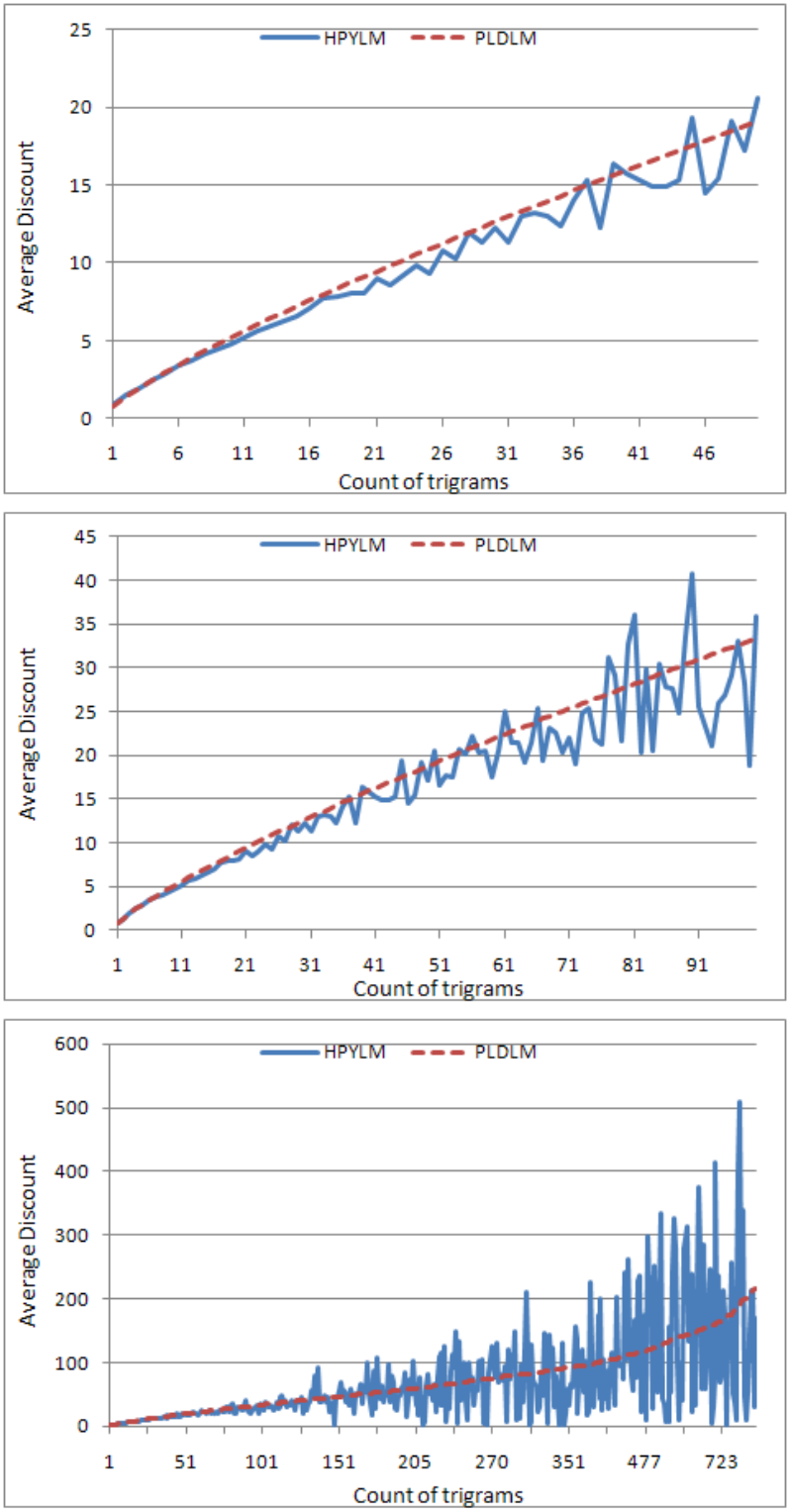


Figure 5.1: Average discounts as a function of trigram counts in the PLDLM and the HPYLM trained on meeting-s1, with different scales for horizontal axis: first 50 counts (top), first 100 counts (middle), and first 1000 counts(bottom).

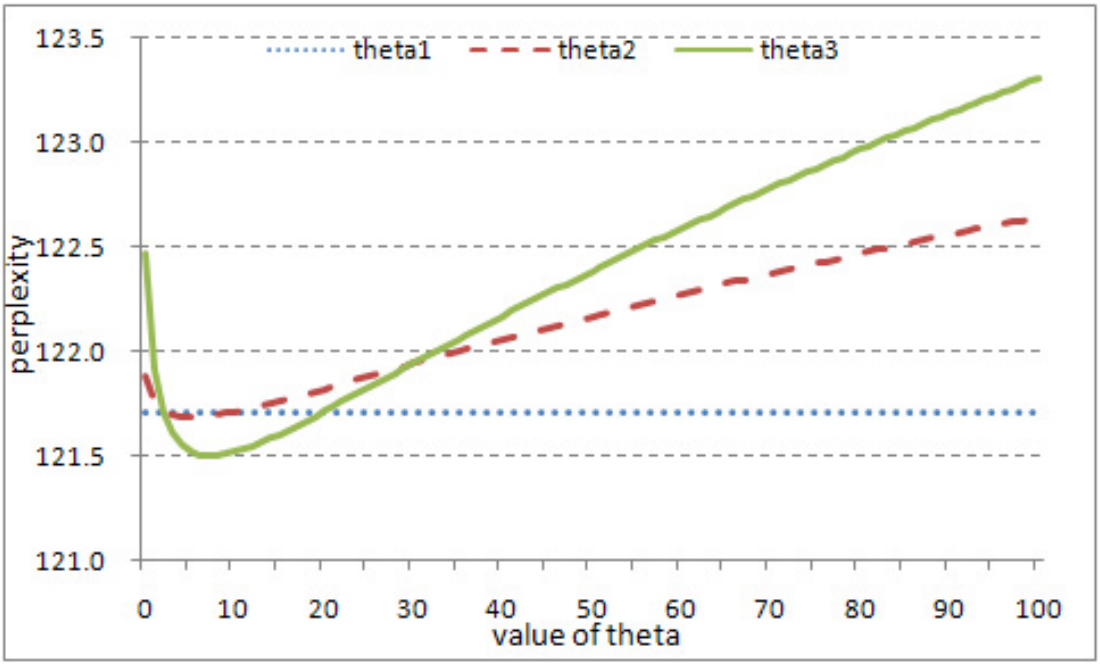


Figure 5.2: Effect of strength parameter θ on *rt06seval*.

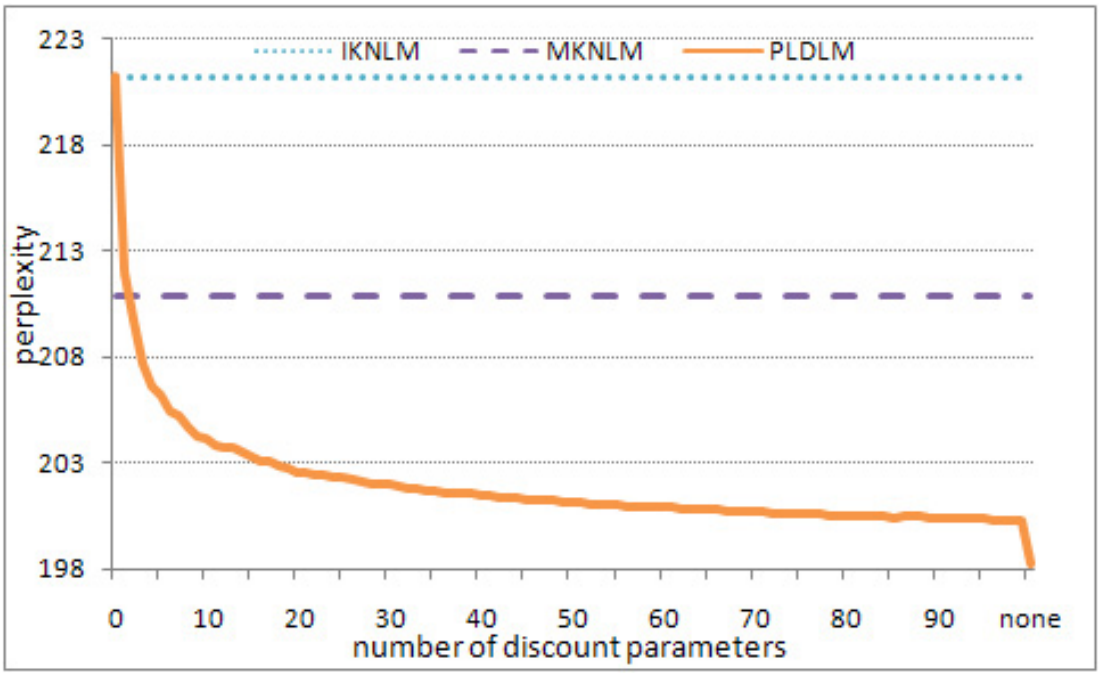


Figure 5.3: Effect of discount parameter numbers on *amieval*.

5.6 Summary

We present in this chapter a simple but efficient smoothing technique for language modelling that makes use of the power law distribution. The PLDLM estimates a smoothing parameter using power law discounting directly, thus avoiding expensive approximate inference, while maintaining comparable computational complexity to the IKNLM and the MKNLM. On the other hand, the PLDLM is an approximation to the HPYLM, producing similar performance to the HPYLM and in turn outperforming the IKNLM and the MKNLM. We conclude that the PLDLM is ready for use in practical large vocabulary speech recognition systems.

Chapter 6

Modelling Topic and Role Information using the HDP

In this chapter, we address the modelling of topic and role information in multiparty meetings, via a nonparametric Bayesian model called the hierarchical Dirichlet process [Teh et al., 2006]. This model provides a powerful solution to topic modelling and a flexible framework for the incorporation of other cues such as speaker role information. We present our modelling framework for topic and role on the AMI Meeting Corpus, and illustrate the effectiveness of the approach in the context of adapting a baseline language model in a large-vocabulary automatic speech recognition system for multiparty meetings. The adapted LM produces significant improvements in terms of both perplexity and word error rate. We also propose the HDPLM in this chapter, which combines the n-gram LM with the HDP model. Finally, we use the HDP model to predict role information.

6.1 Introduction

In this chapter, we investigate language modelling for ASR in multiparty meetings through the inclusion of richer knowledge in a conventional n-gram language model. We use the AMI Meeting Corpus[1] [Carletta, 2007], as described in Section 3.1.1. From the viewpoint of language modelling, the question for us is whether there are cues beyond lexical information which can help to improve an n-gram LM. If so, then what are those cues, and how can we incorporate them into an n-gram LM? To address this question, we here focus on the modelling of topic and role information using a

[1] `http://corpus.amiproject.org`

hierarchical Dirichlet process [Teh et al., 2006].

Consider an augmented n-gram model for ASR, with its context enriched by the inclusion of two cues from meetings: the *topic* and the speaker *role*. Unlike role, which could be seen as deterministic information available in the corpus, topic refers to the semantic context, which is typically extracted by an unsupervised approach. One popular topic model is latent Dirichlet allocation (LDA) [Blei et al., 2003], as introduced in Section 2.3.4.2. LDA can successfully find latent topics based on the co-occurrences of words in a 'document'. However, there are two difficulties arising from the application of LDA to language modelling of multiparty conversational speech. First, it is important to define the notion of document to which the LDA model can be applied: conversational speech consists of sequences of utterances, which do not comprise well-defined documents. Second, it is not easy to decide the number of topics in advance, a requirement for LDA.

The hierarchical Dirichlet process [Teh et al., 2006] is a nonparametric generalisation of LDA which extends the standard LDA model in two ways. First, the HDP uses a Dirichlet process prior for the topic distribution, rather than the Dirichlet distribution used in LDA. This enables the HDP to determine the number of topics required. Second, the hierarchical tree structure enables the HDP to share mixture components (topics) between groups of data. In this chapter we exploit the HDP as our modelling approach for automatic topic learning. Moreover, we also find it easier to incorporate roles together with topics by expressing them as an additional level of variables into the HDP hierarchy.

Some previous work has been done in the area of combining n-gram models and topic models such as LDA and probabilistic latent semantic analysis (pLSA) for ASR on different data, for example, broadcast news [Mrva and Woodland, 2006; Tam and Schultz, 2006], lecture recordings [Hsu and Glass, 2006], and Japanese meetings [Akita et al., 2007]. The new ideas we exploit in this work cover the following aspects. First, we use the nonparametric HDP for topic modelling to adapt n-gram LMs. Second, we consider sequential topic modelling, and define documents for the HDP by placing a moving window over the sequences of short sentences. Third, we incorporate the role information with topic models in a hierarchical Bayesian framework. In the rest of this chapter, we will review topic models, and introduce our framework for modelling topic and role information using the HDP, followed by a set of perplexity and word error rate (WER) experiments.

6.2 Hierarchical Dirichlet Process

LDA uses Dirichlet distributed latent variables to represent shades of memberships to different cluster or topics. In the HDP nonparametric models are used to avoid the need for model selection [Teh et al., 2008]. Two extensions are made in the HDP: first the Dirichlet distributions in LDA are replaced by Dirichlet processes in the HDP as priors for topic proportions; second, the priors are arranged in a tree structure.

6.2.1 Nonparametric Topic Modelling

As introduced in Section 2.3.1.2, the Dirichlet process is a prior widely used for nonparametric modeling. In this section, we will describe how the Dirichlet process is used to extend LDA for nonparametric topic modelling.

Recall in Equation (2.38) for LDA, a finite-dimensional Dirichlet distribution (i.e., in which π is a K-dimensional vector) is used as prior for distribution of topic proportions. LDA, in this sense, is a finite mixture model. If we use an DP instead as prior for mixing topic proportions, that is, $\theta_d \sim \mathrm{DP}(\alpha, H)$ where $\phi_k | H \sim \mathrm{Dir}(\beta\tau)$, then the stick-breaking construction for $\pi \sim \mathrm{GEM}(\alpha)$ will produce a countably infinite dimensional vector π. In this way, the number of topics in this DP-enhanced LDA model is potentially infinite, the number of topics increasing with the available data.

This model, as shown in Figure 6.1.(B), is called the Dirichlet process mixture model (also known as an infinite mixture model).

6.2.2 Hierarchical Framework.

Besides the nonparametric extension of LDA from Dirichlet distribution to Dirichlet process, Teh et al. [Teh et al., 2006] further extended the Dirichlet process mixture model from a flat structure to a hierarchical structure, called a hierarchical Dirichlet process mixture model. This extended model uses the hierarchical Dirichlet process as priors. Similar to the DP, the HDP is a prior for nonparametric Bayesian modelling. The difference is that in the HDP, it is assumed that there are groups of data, and that the infinite mixture components are shared among these groups.

Considering a simple 2-level HDP as an example, as shown is Figure 6.1.(C), the HDP defines a set of random probability measure G_j, one for each group of data, and a global random probability measure G_0. The global measure G_0 is distributed as an DP with concentration parameter γ and base probability measure H, and the random

measure G_j, assuming conditionally independent given G_0, are in turn distributed as an DP with concentration parameter α and base probability measure G_0:

$$G_0|\gamma, H \sim \mathrm{DP}(\gamma, H) \qquad G_j|\alpha, G_0 \sim \mathrm{DP}(\alpha, G_0) \tag{6.1}$$

This results in a hierarchy of DPs, in which their dependencies are specified by arranging them in a tree structure. Although this is a 2-level example, the HDP can readily be extended to as many levels as required.

An HDP-enhanced LDA model, therefore, will have a potentially infinite number of topics, and these topics will be shared among groups of data. If an HDP is used as a prior for topic modelling, then the baseline distribution H provides the prior distribution for words in the vocabulary, i.e., $\phi_k|H \sim \mathrm{Dir}(\beta\tau)$. The distribution G_0 varies around the prior H with the amount of variability controlled by γ, i.e., $G_0 \sim \mathrm{DP}(\gamma, \mathrm{Dir}(\beta\tau))$. The actual distribution G_d for dth group of data (words in dth document in topic models) deviates from G_0, with the amount of variability controlled by α, i.e., $G_d \sim \mathrm{DP}(\alpha, G_0)$. Together with Equation (6.2), this completes the definition of an HDP-enhanced LDA topic model.

6.3 Topic Modelling using the HDP

We revisit probabilistic topic modelling that was introduced in Section 2.3.4. In topic models, each document $d = 1, \ldots, D$ in the corpus is represented as a mixture over latent topics (let θ_d be the mixing proportions over topics), and each topic $k = 1, \ldots, K$ in turn is a multinomial distribution over words in the vocabulary (let ϕ_k be the vector of probabilities for words in topic k). LDA pioneered the use of Dirichlet distribution as the prior for topic distribution θ_d. Figure 6.1(A) depicts the graphical model for LDA. The generative process for words in each document is as follows: first draw a topic k with probability θ_{dk}, then draw a word w with probability ϕ_{kw}. Let w_{id} be the ith word token in document d, z_{id} the corresponding drawn topic, and Dirichlet priors are placed over the parameters θ_d and ϕ_k, then

$$\begin{aligned} z_{id}|\theta_d &\sim \mathrm{Mult}(\theta_d) \quad w_{id}|z_{id}, \phi_{z_{id}} \sim \mathrm{Mult}(\phi_{z_{id}}) \\ \theta_d|\pi &\sim \mathrm{Dir}(\alpha\pi) \qquad \phi_k|\tau \sim \mathrm{Dir}(\beta\tau) \end{aligned} \tag{6.2}$$

where π and τ are the corpus-wide distributions over topics and words respectively, and α and β are called the concentration parameters, controlling the amount of variability from θ_d/ϕ_k to their prior means π/τ.

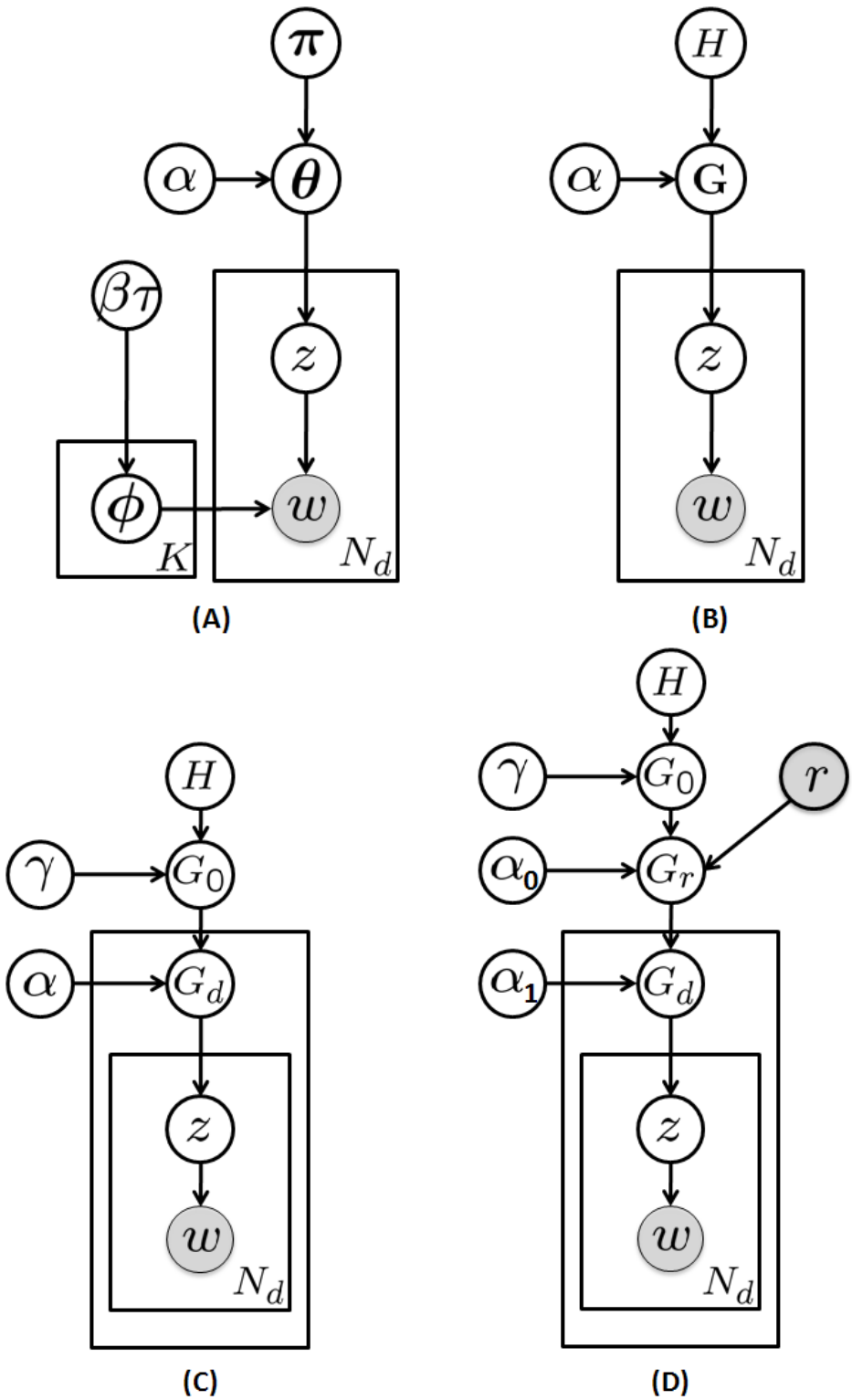


Figure 6.1: Graphical model depictions for (A) latent Dirichlet allocation (finite mixture model), (B) Dirichlet process mixture model (infinite mixture model), (C) 2-level hierarchical Dirichlet process model, and (D) the role-HDP where G_r denotes the DP for one of the four roles (PM, ME, UI, and ID). Each node in the graph represents a random variable, where shading denotes an observed variable. Arrows denote dependencies among variables. Rectangles represent plates, or repeated sub-structures in the model.

In LDA, the number of topics K is determined in advance, i.e., π and θ_d are finite-dimensional vectors. The HDP, on the other hand, is a nonparametric extension to LDA (as described in Section 6.2.1), by using the stick-breaking construction [Teh et al., 2006] for π to accommodate a countably infinite number of topics, i.e., π and θ_d are now both infinite-dimensional vectors:

$$\pi'_k \sim \text{Beta}(1,\gamma) \qquad \pi_k = \pi'_k \prod_{l=1}^{k-1}(1-\pi'_k) \tag{6.3}$$

A random measure defined as $G = \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k}$ is then called a Dirichlet process (DP), with point masses located at ϕ_k. We write $G \sim \text{DP}(\gamma, H)$, with concentration parameter γ, base probability measure H, and $\phi_k|H \sim \text{Dir}(\beta\tau)$. Reformulating topic modelling using the HDP according to [Teh et al., 2006], we have

$$G_0|\gamma, H \sim \text{DP}(\gamma, H) \quad G_d|\alpha, G_0 \sim \text{DP}(\alpha, G_0) \tag{6.4}$$

Figure 6.1(B) shows the corresponding 2-level HDP, which can be readily extended to as many levels as required.

6.3.1 Document Representation

The way we define a *document* in topic models is important, since it affects the scope of word co-occurrences to be considered for topic modelling. We used a moving window to define documents for the HDP: first align all words in a meeting along a common timeline; then for each sentence/segment, backtrace and collect those non-stop words belonging to a window of length L beginning from the end time of the sentence/segment.

The target application of the HDP in this thesis is the adaptation of LMs for a multiparty conversational ASR system. For each sentence in the testing data, we need to find a corresponding document for the HDP, based on which topics are extracted, and then the LM is dynamically adapted according to the topic information. Documents also include information about speaker role. In the AMI Meeting Corpus, meetings are manually annotated with word transcription (in `*.words.xml`), with time information being further obtained via forced alignment. Also available in the corpus are the segment annotations (in `*.segments.xml`). Role information can be easily determined from the annotations in the corpus. We used the following procedure, as shown in Algorithm 4, to obtain documents: for each scenario meeting, first align all the words in it along a common timeline; then for each sentence/segment, collect those non-stop

words belonging to a window of length L, by backtracing from the end time of the sentence/segment, as the document. The role that has been assigned to the most of words in the window is selected as the role for that document.

Algorithm 4: EXTRACTDOCUMENTS($\mathbb{C}$): an procedure for the definition of documents for the HDP and roleHDP.

```
procedure EXTRACTDOCUMENTS( ℂ )
input : meeting corpus ℂ
for each meeting m in the corpus ℂ do
    retrieve words with time and role information for m
    align all words in m to a common timeline
    for each segment s of meeting m do
        st = STARTTIME(s)
        et = ENDTIME(s)
        if et-st < window length L then
            st = et - L
        endif
        for each word w appearing within segment words[st:et] do
            if NONSTOPWORD(w) then
                document_s += w
            endif
        endfor
        role_s = role assigned to most words;
    endfor
endfor
```

By collecting all documents for meetings belonging to the training and testing data respectively, we can obtain the training data for HDP model and the testing data for perplexity evaluation. A similar idea applies to finding documents dynamically for ASR experiments. The difference is that we do not have the segment annotations in this case. Instead speech segments, obtained by either automatic or manual approaches, are used as units for finding documents as well as for ASR. In the ASR case we use an online unsupervised method: ASR hypotheses (with errors and time information) from previous segments are used to define documents for HDP inference for the current segment. In both cases above, we simply ignore those segments without corresponding

documents.

As a comparison, we will present results in Section 6.7 for topic models trained using different document representations, for example, the manually annotated segments, and the short-span n-gram contexts. There may not be much difference between the final topics using different document representation. However, the way we use here provides a better approach for us to carry out the dynamic adaptation of language models.

6.3.2 Inference

We will briefly introduce the inference algorithm we use for the HDP. More specifically, we follow the sampling-based inference method in [Teh et al., 2006; Teh and Jordan, 2010]. Each training document D_j in the corpus corresponds to a Dirichlet process G_j, which shares a global Dirichlet process G_0. According to Chinese restaurant representation, the first task of the sampler is to sample the tables z_{ji} where customers x_{ji} sit in the restaurant j, according to the following probibilities:

$$z_{ji} = \begin{cases} k & \text{with probability} \propto (n_{jk}^{-ji} + \alpha\beta_k) f_k^{-x_{ji}}(x_{ji}) \\ k^{\text{new}} & \text{with probability} \propto \alpha\beta_0 f_{k^{\text{new}}}^{-x_{ji}}(x_{ji}) \end{cases} \tag{6.5}$$

If a new table k^{new} is required, a new topic is initialized and added to the model. The corresponding weights for the new topic are given by:

$$\begin{aligned} v_0|\gamma &\sim \text{Beta}(\gamma, 1) \\ (\beta_0^{\text{new}}, \beta_{K+1}^{\text{new}}) &= (\beta_0 v_0, \beta_0(1-v_0)) \\ v_j|\alpha, \beta_0, v_0 &\sim \text{Beta}(\alpha\beta_0 v_0, \alpha\beta_0(1-v_0)) \\ (\pi_{j0}^{\text{new}}, \pi_{jK+1}^{\text{new}}) &= (\pi_{j0} v_j, \pi_{j0}(1-v_j)) \end{aligned} \tag{6.6}$$

After sampling the seating arrangement z_{ji} in restaurant j, the stick-breaking weights for G_0 can be sampled conditioned on z_{ji}:

$$\beta_0, \beta_1, \ldots, \beta_K | \gamma, G_0 \sim \text{Dirichlet}(\gamma, m_{\bullet 1}, \ldots, m_{\bullet K}) \tag{6.7}$$

Similarly, the stick-breaking weights for G_j is giving by:

$$\pi_{j0}, \pi_{j1}, \ldots, \pi_{jK} | \alpha, G_j \sim \text{Dirichlet}(\alpha\beta_0, \alpha\beta_1 + n_{j\bullet 1}, \ldots, \alpha\beta_K + n_{j\bullet K}) \tag{6.8}$$

In summary, the inference algorithm for an HDP is shown in Algorithm 5.

Algorithm 5: INFERHDP(): an algorithm for the inference of an HDP, where K is the current number of topics.

```
procedure INFERHDP( order n )
for each restaurant j do
    for each each customer i in restaurant j do
        sample the table z_ji where customer x_ji sits according to Equation (6.5)
        if z_ji=k^new then
            initialize and add a new topic according to Equation (6.6)
            z_ji = K + 1
            K = K + 1
        endif
    endfor
    sample β according to Equation (6.7)
    sample π_j according to Equation (6.8)
endfor
sample concentration parameters γ and α
```

There are other inference algorithms that are based on variational methods, for example, [Liang et al., 2007; Teh et al., 2008].

6.3.3 Combination with n-gram LMs

A topic in an HDP is a multinomial distribution over words in the vocabulary (denoted as ϕ_k), which can be considered as a unigram model. To be precise, we use $P_{\text{hdp}}(w|d)$ to denote the unigram probabilities obtained by the HDP based on the jth document d. The HDP probability $P_{\text{hdp}}(w|d)$ is approximated as a sum over all the latent topics ϕ_k for that document, supposing there are K topics in total in the HDP at the current time:

$$P_{\text{hdp}}(w|d) \approx \sum_{k=1}^{K} \phi_{kw} \cdot \theta_{dk} \tag{6.9}$$

where the probability vector ϕ_k is estimated during training and remains fixed in testing, while the topic weights $\theta_d|G_0 \sim \text{DP}(\alpha_0, G_0)$ are document-dependent and thus are calculated dynamically for each document. For the roleHDP, the difference is that the topic weights are derived from role DPs, i.e., $\theta_d|G_{\text{role}} \sim \text{DP}(\alpha_1, G_{\text{role}})$.

One simple way to combine the topic model $P_{\text{hdp}}(w|d)$ and traditional n-gram language model $P_{\text{back}}(w|h)$ is to linearly interpolate the two models using different weights λ.

$$P_{\text{interp}}(w|h) = \lambda_1 * P_{\text{back}}(w|h) + \lambda_2 * P_{\text{hdp}}(w|d) \tag{6.10}$$

Another approach to the combination takes the probability from topic model as the dynamic marginal to dynamically adapt the baseline language model. As in [Kneser et al., 1997], we treat $P_{\text{hdp}}(w|d)$ as a dynamic marginal and use the following equation to adapt the baseline n-gram model $P_{\text{back}}(w|h)$ to get an adapted n-gram $P_{\text{adapt}}(w|h)$, where $z(h)$ is a normalization factor, and μ is a scale factor:

$$P_{\text{adapt}}(w|h) = \frac{\alpha(w)}{z(h)} \cdot P_{\text{back}}(w|h) \qquad \alpha(w) \approx \left(\frac{P_{\text{hdp}}(w|d)}{P_{\text{back}}(w)}\right)^{\mu} \tag{6.11}$$

Since we require to calculate the normalization factor $z(h)$ for each of the context h, the adaptation approach in Equation 6.11 could be more computationally expensive than the interpolation method in Equation 6.10. Generally, the computing time grows with respect to the number of words in the vocabulary, and the number of contexts occurring in the training corpus.

The value of scale factor μ can be obtained by tuning it on a development set. Alternatively, we can empirically set the value of μ, for example, we set $\mu = 0.5$ in the experiments in Section 6.7.

6.3.4 Vocabulary Mismatch

There are 50k words in the vocabulary V_{asr} of our baseline LMs. After removing stop words in the AMI meeting corpus, we fix the size of vocabulary V_{hdp} as 7,910 words for our HDP/roleHDP models. We get zero probabilities for $P_{\text{hdp}}(w|d)$ in Equation (6.11) for those $w \notin V_{\text{asr}}$, which will be problematic for N-best rescoring. Therefore, we deal with this vocabulary mismatch problem in two ways: 1) *model interpolation*, in which for those $w \notin V_{\text{asr}}$ we directly assign the unigram probabilities from the background LMs to $P_{\text{hdp}}(w|d)$; and 2) *count interpolation*, in which $P_{\text{hdp}}(w|d) = \frac{C_{\text{back}}(w)+C_{\text{hdp}}(w)}{\sum_{w'}(C_{\text{back}}(w')+C_{\text{hdp}}(w'))}$ for each $w \in V_{\text{asr}}$. The second method corresponds to the MAP adaptation from the background unigram LMs for each topic by interpolating the count statistics from the background unigram $C_{\text{back}}(w)$ and the HDP $C_{\text{hdp}}(w)$ (normally boosted by some weights).

6.4 Hierarchical Dirichlet Process LM

In this section, we present our study on more tightly combining topic models and n-gram LMs. More specifically, we try to extend the hierarchical Dirichlet process topic models to take into consideration the word order information. This is attractive, because topic models and language models can benefit from each other when tightly combined: by considering word order, topic models are expected to extract more reasonable topics; and by considering topics, language models can in turn rely on richer knowledge for word prediction.

6.4.1 Motivation

So far, we have introduced two approaches to topic modelling, latent Dirichlet allocation (LDA) in Chapter 2, and the hierarchical Dirichlet process (HDP) in Chapter 6. One assumption made by these topic models is that they are "bag-of-words" models, i.e., word order is ignored in these models. We have shown in Chapter 6 how we can estimate a unigram LM from the HDP topic model. This, however, leaves a question of how to combine topic models with the traditional n-gram models. One straightforward method to do the combination is to use linear interpolation, by simply interpolating the unigram LM from the topic model and the baseline n-gram LM. This method, however, does not work well in practice. A better way to do the combination is to use the unigram from the HDP as the dynamic marginal to dynamically adapt the baseline n-gram LM, as shown in Equation (6.11) in Section 6.3.3. This approach suffers from the need for expensive computation to compute the normalization term for each context in Equation (6.11). Therefore, we normally have to employ this combination approach for N-Best or lattice rescoring, which limits its application in a large vocabulary ASR system.

Since word order is often critical to capturing the meaning of text in practice, and there has been previous research on incorporating word order information into topic models. The HMM-LDA model [Griffiths et al., 2004], which models both short-span syntactic dependencies and long-span semantic dependencies between words. A hidden Markov model was used to capture the syntactic information, and latent Dirichlet allocation was used to model the semantic knowledge. This is the first attempt to introduce word dependency to topic modelling. Wallach [2006] combined a bigram hierarchical Dirichlet LM (HDLM) and an LDA topic model within the hierarchical Bayesian framework, and observed some reductions in perplexity. Wang et al.

[2007] presented topical n-grams, a topic model that discovers topics as well as topical phrases. To generate words in their textual order, the probabilistic model first samples a topic, then samples its status as a unigram or a bigram, and then samples the word from a topic-specific unigram or bigram distribution. The model is thus able to distinguish the phrase "white house" from either a "politics" topic or a "real estate" topic. In this chapter, we follow the work in [Wallach, 2006] by extending it to the n-gram case, and using the HDP for topic modelling, which we term the hierarchical Dirichlet process language model (HDPLM).

6.4.2 The Model

We first briefly introduce the bigram topic model in [Wallach, 2006], which combines the bigram HDLM with LDA. The bigram HDLM basically uses an additive smoothing for the empirical probability ($c_w/c_\bullet$), being smoothed by the hyperparameter from the prior Dirichlet distribution, as shown in Equation (4.2). We reformulate the predictive distribution in Equation (4.2) using a simpler notation, $w_t = i, w_{t-1} = j$, in Equation (6.12).

$$P(i|j,\beta\mathbf{m}) = \frac{c_{i|j}}{c_j+\beta} + \frac{\beta m_i}{c_j+\beta} \tag{6.12}$$

where $c_{i|j}$ is the number of occurrences of $j|i$ and $c_j = \sum_i c_{i|j}$. β is the hyperparameter of the Dirichlet distribution in the HDLM and $\beta > 0$, and $\mathbf{m}$ is a measure satisfying $\sum_i m_i = 1$. Since LDA uses the same prior distribution as the HDLM, the word distribution given a topic k in LDA has the similar form as Equation (6.12), except that it is indeed a unigram given each topic.

$$P(i|k,\beta\mathbf{m}) = \frac{c_{i|k}}{c_k+\beta} + \frac{\beta m_i}{c_k+\beta} \tag{6.13}$$

where k represents a topic in LDA, and $c_k = \sum_i c_{i|k}$ is the total number of words occurring in topic k.

[Wallach, 2006] extended latent Dirichlet allocation by incorporating a notation of word order. Word generation in this model is defined by a conditional distribution $P(w_t = i|w_{t-1} = j, z_t = k)$ described by $|V|K(|V|-1)$ parameters, where z_t is the topic assignment for time t, $|V|$ is size of the vocabulary, and K is the total number of topics. Each topic k is now characterized by the $|V|$ distributions specific to that topic. In this way, Wallach [2006] combined Equation (6.12) and Equation (6.13), and reached a predictive distribution shown in Equation (6.14).

$$P(i|j,k,\beta\mathbf{m}) = \frac{c_{i|j,k}}{c_{j,k}+\beta} + \frac{\beta m_i}{c_{j,k}+\beta} \tag{6.14}$$

We can either use a global set of hyperparameters β and $\mathbf{m}$, as in Equation (6.14), or use topic specific hyperparameters, i.e., β_k and $\mathbf{m}_k$, as in Equation (6.14).

$$P(i|j,k,\beta_k\mathbf{m}_k) = \frac{c_{i|j,k}}{c_{j,k}+\beta_k} + \frac{\beta_k m_{i|k}}{c_{j,k}+\beta_k} \tag{6.15}$$

Comparing to Equation (6.12) and Equation (6.13), we can interpret Equation (6.14) or Equation (6.15) in two different perspectives, either that it factorizes the conditional distribution in Equation (6.12) further into K topics, or that it factorizes the conditional distribution in Equation (6.13) further $|V|$ contextual words. Therefore, *factorization* is the key idea of the model in [Wallach, 2006].

We propose a hierarchical Dirichlet process LM, an extension to the "beyond bag-of-words" model in [Wallach, 2006]. The main ideas and extensions of the HDPLM are as follows:

- Instead of using a bigram LM, we provide a flexible framework to extend to n-gram LMs.
- Instead of using LDA for topic modelling where we require to specify the number of topics beforehand, we use the HDP for topic modelling, without the need of specifying the number of final topics to model, and extend it to take into consideration word order information.
- Instead of using an EM algorithm to estimate the hyperparameters in [Wallach, 2006], we use MCMC Gibbs sampling to infer the hyperparameters for the HDP.
- Instead of relying on topic inference during testing, we estimate a smoothed ARPA format n-gram LM, making it flexible for use in a practical large vocabulary ASR system.

Let the contexts in the HDPLM be represented by $\mathbf{u} = h$. We describe the main steps to estimate an HDPLM in the following:

1. Without removing the stop words from the training data, read and organise the data into a *trie* structure, as the same data structure as for traditional n-gram LMs. Each context h has a Dirichlet process associated with it, as shown in Figure 6.2 the example of a trigram LM.
2. The observation data – document – for each DP is composed by those words following the specific context h. That is, we sort the training data with respect

to different n-gram contexts, and the HDP is used to model the co-occurrence between those words that follow the same context h. Unlike the document representation in traditional topic models, the documents defined for the HDPLM here may be short, i.e., less than ten words in a document.

3. Initialize the HDP model represented in Figure 6.2, and do the MCMC Gibbs sampling for sufficiently enough iterations to burn-in. Keep track of the factorization statistics for each context h during the inference, i.e., the factorization of the $c_{i|h}$ occurrence of word i into K topics $c_{i|h,k}$.

4. After the HDP model is converged, estimate the predictive distribution for each context h using Equation (6.16), where the hyperparameters $\beta_{h,k}$ and $\mathbf{m}_{h,k}$ are specific to each context h and topic k, and are tied together in the hierarchy of the HDP framework.

$$P(i|h,k,\beta_{h,k}\mathbf{m}_{h,k}) = \frac{c_{i|h,k}}{c_{h,k}+\beta_{h,k}} + \frac{\beta_{h,k}m_{i|h,k}}{c_{h,k}+\beta_{h,k}} \tag{6.16}$$

5. By integrating out all K topics in Equation (6.16), we obtain the predictive distribution $P(i|h)$ for each context h, and estimate a final APRA-format n-gram LM.

The predictive distribution given in Equation (6.16) is similar to that in Equation (6.15) by [Wallach, 2006], except the fact that the former extends to n-gram case while the latter uses a bigram LM, and the fact that the former uses the HDP model to automatically determine the number of final topics.

If we manually force the number of topics to be one, i.e., $K = 1$, then the HDPLM in Equation (6.16) will revert to the HDLM in Equation (6.12). Similarly, if we manually force the order of the HDPLM to be one, i.e., $n = 1$, then the HDPLM in Equation (6.16) will revert to LDA model in Equation (6.13).

There are two issues to remark on for the HDPLM. First, the documents used for topic modelling in the HDPLM are normally shorter than usual – each document consists of only those words following a specific context h. Second, there is no advanced smoothing technique such as discounting applied in Equation (6.16), except for the additive smoothing. This means that the HDPLM is incapable of producing comparable results to state-of-the-art smoothed LMs, for example, the IKNLM and the MKNLM.

Despite lack of advanced smoothing techniques in the HDPLM, it is still interesting to investigate the HDPLM because of the following two reasons. Firstly, the HDPLM

can be regarded as a type of aspect model [Hofmann and Puzicha, 1998] in which the predictive probability of the HDPLM is factorized by automatically derived topics. Each topics in the HDPLM corresponds to one aspect. The HDPLM is therefore superior to the additive smoothing. We will see from perplexity results in Table 6.5 the advantage arising from introducing more aspects in the HDPLM. Secondly, we can apply advanced smoothing techniques such as Kneser-Ney smoothing on each individual aspect of the HDPLM, and expect the HDPLM to product better performance than the state-of-the-art smoothed LMs.

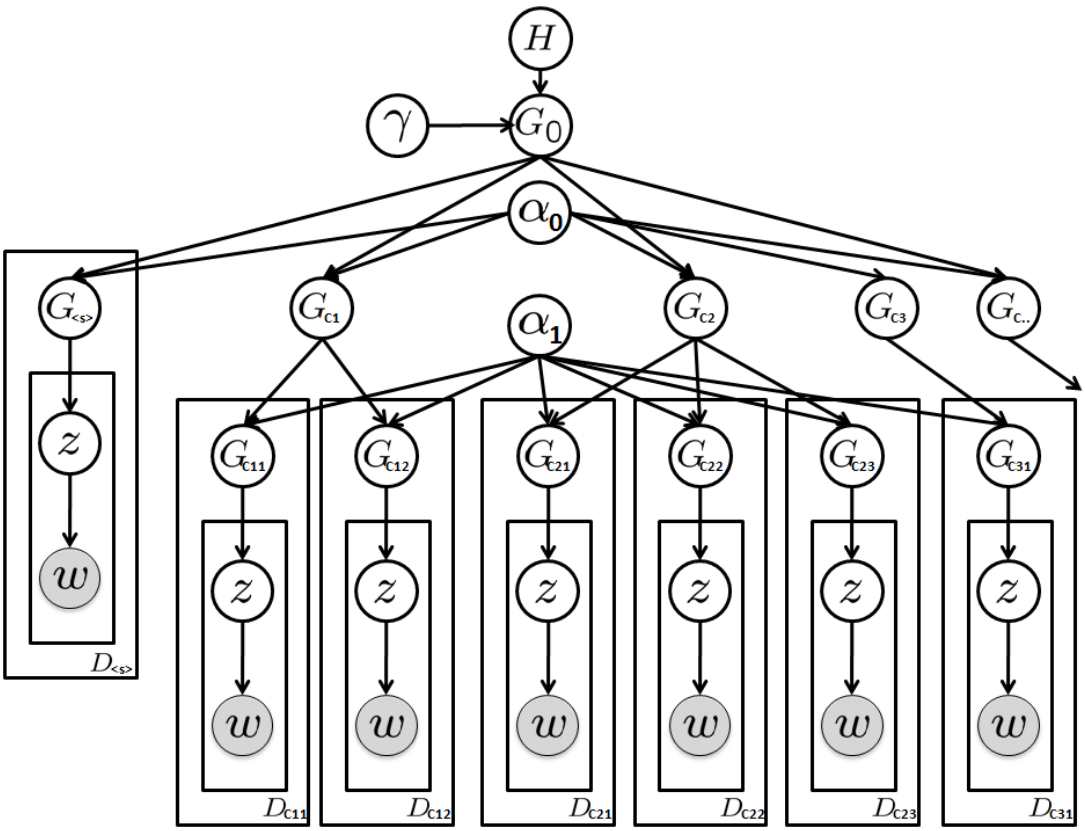


Figure 6.2: The graphical model representation for a trigram HDPLM, which combines the HDP and the n-gram LM. Each context of the HDPLM has an DP associated with it, and these DPs are organised into a hierarchical structure to share information. The observation data of a leaf DP is composed by those words following the corresponding context.

6.5 Incorporating Role Information into the HDP

6.5.1 Factored Language Model

One straightforward method for modelling words and roles is to use the maximum likelihood estimation based on the co-occurrences of words w and the role information r, i.e., training a bigram-like model $P(w|r) = \text{Count}(w,r)/\text{Count}(r)$. More generally, we can use a factored language model [Bilmes and Kirchhoff, 2003] to model words and role deterministically. The FLM, initially developed to address the language modelling problems faced by morphologically rich or inflected languages, is a generalisation of standard n-gram language models, in which each word w_t is decomposed into a bundle of K word-related features (called *factors*), $w_t \equiv f_t^{1:K} = \{f_t^1, f_t^1, \ldots, f_t^K\}$. Factors may include the word itself. Each word in an FLM is dependent not only on a single stream of its preceding words, but also on additional parallel streams of factors. Combining with interpolation or generalized parallel backoff (GPB) [Bilmes and Kirchhoff, 2003] strategies, multiple backoff paths may be used simultaneously.

We exploit two factors for word w at time t: the word w_t itself and the corresponding role r_t, as shown in Figure 6.3. All the words in a sentence share a common role, i.e., $r_t = r_{t-1} = \ldots = r_1$ in Figure 6.3. We use a simple backoff strategy, for example, by moving from the model $P(w_t|r_t)$ directly down to the unigram model $P(w_t)$. We refer this model to the role-FLM.

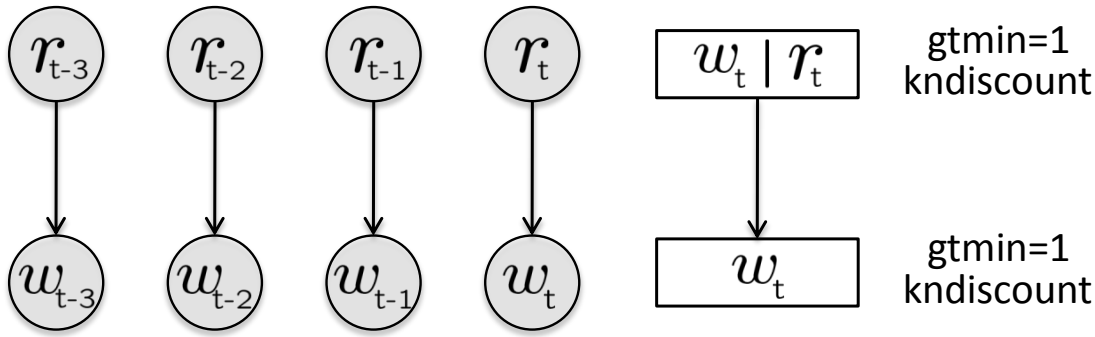


Figure 6.3: The graphical model representation and backoff path for role-FLM.

6.5.2 HDP for Role Information (roleHDP)

We consider the problem of introducing role into the HDP hierarchy to enable better topic modelling. In the scenario meetings of the AMI Meeting Corpus, each of the four participants in a meeting series was assigned a different role (PM, ME, UI, or ID).

Since different participants have different roles to play, there may be a different topic distribution, and in turn different dominant words, specific to each role. However, we still expect topic models to work as a whole on the corpus rather than having four separate topic models. The HDP is thus an appropriate model, because it has a flexible framework to express DP dependencies using a tree structure.

An alternative approach to modelling the relationship between the participant role information and lexical words is to directly estimate the conditional probability $P(w|r)$ based on the co-occurrence statistics of roles and words, using the maximum likelihood principle. As a comparison to the probabilistic topic models, we also introduce in this section a deterministic approach to modelling roles and words, by regarding the role as an additional feature (factor) of lexical words in an MLE-based LM.

Documents were defined as described above for those scenario meetings with role information, a one-to-one mapping. We grouped the documents for each of the four roles, and assigned an DP G_{role} for each role, which then served as the parent DP in the HDP hierarchy (the base probability measure) for all DPs corresponding to documents belonging to that role. To share the topics among four roles, a global DP G_0 was used as the common base probability measure for the four role DPs G_{role}. See the graphical model shown in Figure 6.1.(C) for the HDP hierarchy. Formally speaking, we used a 3-level HDP, referred to as roleHDP, to model topic and role information in the AMI Meeting Corpus:

$$G_0|\gamma, H \sim \text{DP}(\gamma, H), G_{\text{role}}|\alpha_0, G_0 \sim \text{DP}(\alpha_0, G_0), G_j|\alpha_1, G_{\text{role}} \sim \text{DP}(\alpha_1, G_{\text{role}}) \quad (6.17)$$

6.6 Predicting Role Information using the roleHDP

In Section 6.5.2, we make use of role information for topic modelling by treating the role as an *observed* variable in the hierarchical Dirichlet process model during testing, as the graphical model representation shown in Figure 6.4. In this section, we treat the role in an opposite way, by making the role as a *latent* variable in the HDP framework during testing, and trying to predict the participant's role $\hat{r}$ in the AMI Meeting Corpus, one of project manager (PM), marketing expert (ME), user interface designer (UI) and industrial designer (ID), based on a single short document D_{test}, i.e., several sentences. For example, we expect the model to predict the role of the following document/sentences [2] to be a project manager (PM):

[2]The example document is taken from the testing data, exactly as presented to human annotators.

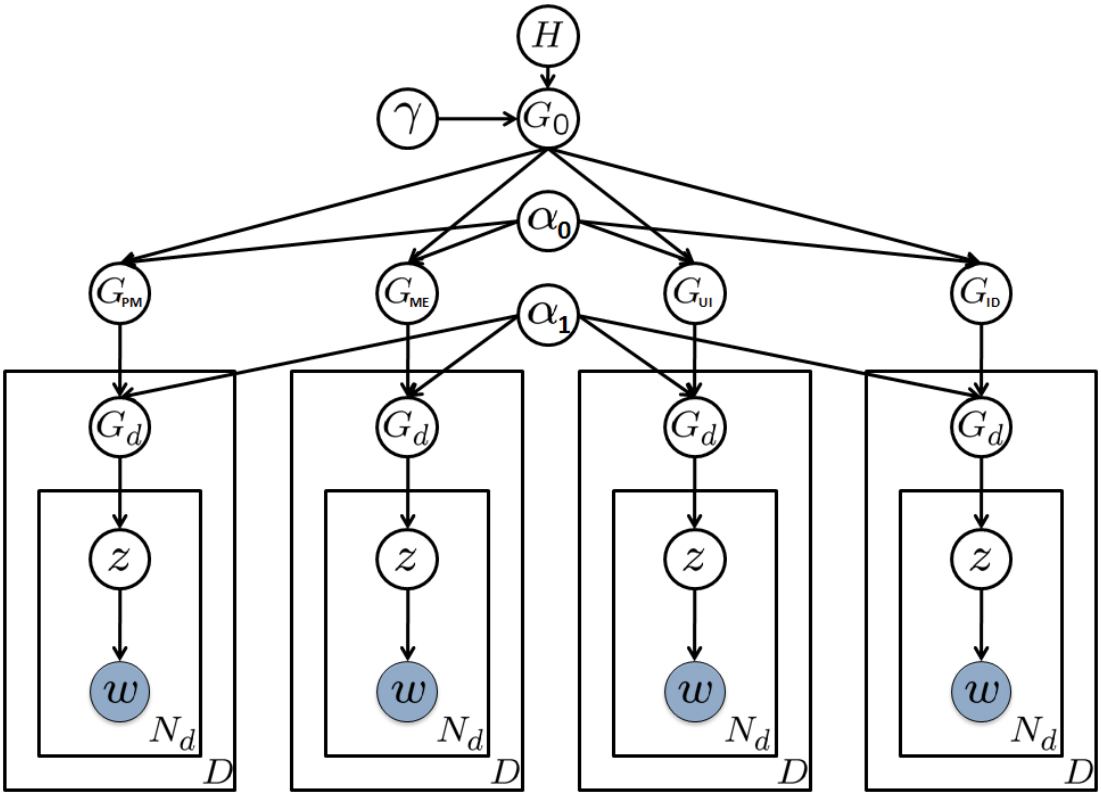


Figure 6.4: The unfolded graphical model representation for role-HDP.

```
START OF THE FIRST MEETING UH
RIGHT SO AGENDA OF THE FIRST MEETING WHERE WE UH
WE HAVE TWENTY FIVE MINUTES FOR THIS MEETING WE UH ARE TO GET
ACQUAINTED SO DOES EVERYONE WANT TO SAY WHO THEY ARE THAT
SEEM SENSIBLE
```

Although this task is not a real application for multiparty meetings, it could be helpful for some tasks in the scenario meetings, for example, automatic camera selection.

One straightforward way to do this is to use n-gram LMs. We can train four different n-gram language models, each using only those data belonging to the role. For the testing, we select the role whose LM has the highest predictive likelihood on the testing sentences D_{test}:

$$\hat{r} = \arg\max_{r} P_r^{(\text{LM})}(D_{\text{test}}) \qquad r \in \{\text{PM, ME, UI, ID}\} \tag{6.18}$$

We can use any order of n-gram LMs to compute $P_r^{(\text{LM})}(D_{\text{test}})$. Here we try both unigram and trigram LMs.

Alternatively, we may use topic models, either using LDA or the HDP, to classify the test document into one of the four roles. We train four topic models, one for each role. Each using only those training data belong to the role. Similarly to Equation (6.18), we select the role whose topic model (unigram) has the highest predictive likelihood on the testing sentences D_{test}:

$$\hat{r} = \arg\max_{r} P_r^{(\text{TM})}(D_{\text{test}}) \qquad r \in \{\text{PM, ME, UI, ID}\} \tag{6.19}$$

In the case of using the HDP for topic modelling, we compute $P_r^{(\text{TM})}(D_{\text{test}})$ for document $d = D_{\text{test}}$ using Equation (6.20).

$$P_r^{(\text{TM})}(D_{\text{test}}) = P_r^{(\text{HDP})}(D_{\text{test}}) = \prod_{w \in D_{\text{test}}} \sum_{k=1}^{K} \phi_{kw} \cdot \theta_{dk} \tag{6.20}$$

where K is the number of topics in the HDP model, θ_{dk} is the weight for topic k, and ϕ_{kw} is the word distribution for topic k.

We propose to predict roles in the roleHDP framework, as shown in Figure 6.5, and derive a sampling-based approach to predict roles using the roleHDP model. The steps of this approach are as follows:

1. train the roleHDP model on the training data, as describe in Section 6.5.2;
2. associate with each testing document D_{test} a vector of integers, $V^{(D_{\text{test}})} = [N_{\text{PM}}, N_{\text{ME}}, N_{\text{UI}}, N_{\text{ID}}]$, whose elements keep track of the accumulated number of assigned roles. For example, N_{PM} records the number of times that the document has been assigned the role of an PM, and so on. The vector is initialized with a uniform distribution, i.e., $V^{(D_{\text{test}})} = [1, 1, 1, 1]$.
3. do the following loop until convergence:
   (a) for each testing document D_{test} in the test set:
      i. sample the role for document D_{test} according to the following 4-dimensional multinomial distribution $\mathbf{v} = [v_{\text{PM}}, v_{\text{ME}}, v_{\text{UI}}, v_{\text{ID}}]$ which is a function of $V^{(D_{\text{test}})}$:

$$r^* | V^{(D_{\text{test}})} \sim \text{Mult}(\mathbf{v}) \tag{6.21}$$

$$v_r = \prod_{w \in D_{\text{test}}} \sum_{k=1}^{K} \phi_{kw} \cdot \theta_{G_r k} + \alpha_1 * \frac{N_r^{(D_{\text{test}})}}{\sum_{r'} N_{r'}^{(D_{\text{test}})}} \tag{6.22}$$

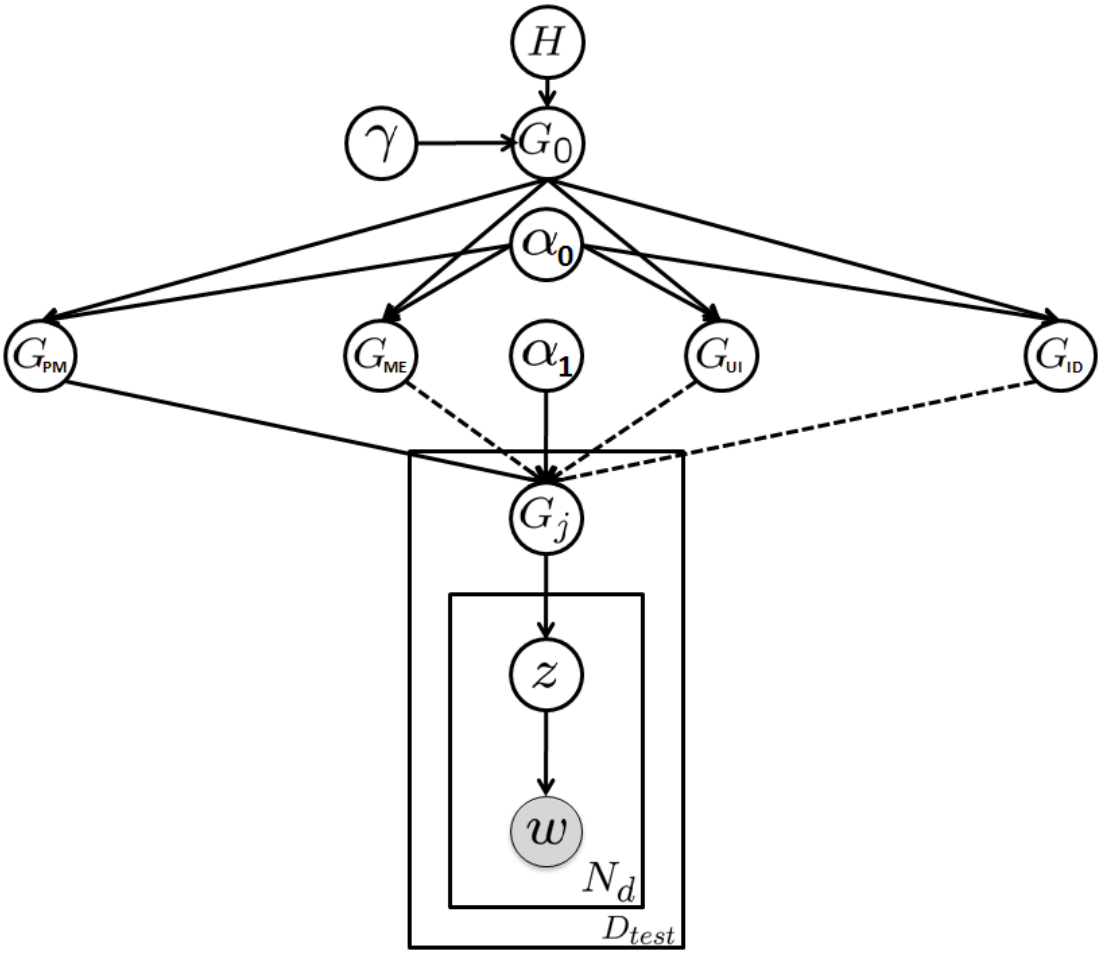


Figure 6.5: The graphical model representation for predicting roles in a roleHDP model. The testing sentences/document is D_{test}. We sample the parent role for G_j according to Equation (6.22). Dotted lines show the potential parent nodes, and solid line is the sampled one.

where $r \in \{\text{PM, ME, UI, ID}\}$. The first term calculates the likelihood of test data given each potential role r – the topic weights $\theta_{G_r k}$ are derived from the DP for role r (G_r, the parent node of $G_{D_{\text{test}}}$ in Figure 6.5) – which is different from that in Equation (6.20). The second term represents the accumulated knowledge. α_1, as shown in Figure 6.5, is the concentrate parameter for the DP $G_{D_{\text{test}}}$.

ii. place the document D_{test} under the Dirichlet process corresponding to role r^*, i.e., change the parent node of the DP for D_{test} to G_{r^*}.

iii. update document-specific vector $V^{(D_{\text{test}})}$, i.e., increasing the number for N_{r^*}.

(b) carry out one sampling iteration for the updated roleHDP, similar to that in

Section 6.5.2.

4. after convergence, assign to each testing document D_{test} the role r which has the maximal value in Equation (6.22). As an approximation, we select the role with the largest value in $V^{(D_{\text{test}})}$ for D_{test}.

If role for the proceeding document of D_{test} is available, this context information may be useful in role prediction. Suppose r_p is the role of the proceeding document of D_{test}, we can train a bigram role LM, and incorporate the conditional role probability $P(r|r_p)$ in Equation (6.22) during sampling:

$$v_r = \prod_{w \in D_{\text{test}}} \sum_{k=1}^{K} \phi_{kw} \cdot \theta_{G_r k} + \alpha_1 * \frac{N_r^{(D_{\text{test}})}}{\sum_{r'} N_{r'}^{(D_{\text{test}})}} + P(r|r_p) \tag{6.23}$$

6.7 Experiments and Results

We report some experimental results in this section. The HDP was implemented as an extension to the SRILM toolkit[3]. All baseline LMs used here were trained using SRILM, and the N-best generation and rescoring were based on a modified tool from SRILM.

6.7.1 Data

Since we considered the role information, which is only available in scenario AMI meetings, we used part of the AMI Meeting Corpus for our experiments. There are 138 scenario meetings in total, and we use two different division of the AMI scenario meeting. In the first division, 118 were used for training and the other 20 for testing (about 11 hours), as shown in Table 6.1. The training set includes both meetings annotated as train and development in the corpus. This is a commonly agreed partition of AMI meetings for evaluation. We used the algorithm introduced in Section 6.3.1 to extract the corresponding document for each utterance. The average number of words in the resulting documents for window lengths of 10 and 20 seconds was 10 and 14 respectively. Data for n-gram LMs were obtained as usual for training and testing. Table 6.1 shows the detailed statistics for data for n-gram models and HDP/roleHDP models.

[3]http://www.speech.sri.com/projects/srilm

Table 6.1: The training and testing meeting data used for experiments in this chapter.

Dataset	#Sent/Doc	#Token	#AvgToken/Sent
Data for n-gram models			
Train	29,450	518,248	17
Test	4,859	94,964	19
HDP/roleHDP, window length = 10 seconds			
Train	19,842	202,634	10
Test	3,331	35,599	10
HDP/roleHDP, window length = 20 seconds			
Train	27,534	394,439	14
Test	4,476	64,616	14

In the second division, we use a 5-fold cross validation setup. The details about the five folds are shown in Table 4.2.

We initialized both HDP and roleHDP models with 50 topics, and $\beta = 0.5$ for Equation (2.38). HDP/roleHDP models were trained on documents of 10 seconds window length from the scenario AMI meetings with a fixed size vocabulary of 7,910 words, using a Markov Chain Monte Carlo (MCMC) sampling method. The concentration parameters were sampled using the auxiliary variable sample scheme in [Teh et al., 2006]. We used 3,000 iterations to 'burn-in' the HDP/roleHDP models.

6.7.2 Empirical Analysis

To empirically analyse the properties and behaviours of nonparametric models, we trained a set of HDP/roleHDP models using various different parameters, for example, the initial number of topics ($k = 1, \ldots, 100$), the prior Dirichlet parameter for topics ($\beta = 0.5, 1.0$ in Equation (2.39), and $\tau_w = 1/W$), and the length of document window ($L = 10, 20$ seconds). For LDA, the symmetric Dirichlet with parameters α_0/K was used for topic distribution θ_d. All models were trained using folds 2–4 of the AMI scenario meetings, with a fixed size vocabulary of 7,910 words, with the Markov Chain

Monte Carlo (MCMC) sampling method. The concentration parameters were sampled using the auxiliary variable sample scheme in [Teh et al., 2006]. We ran 3,000 iterations to burn-in, then collected 10 samples from the posteriors to calculate the unigram perplexity on the fold-1 testing data, with a sample step of 5. Figure 6.6 and 6.7 shows the results, in which some random effects exist because they were based on only one run. We are interested in the following questions:

6.7.2.1 The comparison to LDA.

The best number of topics K for LDA is in the range 10–20. With appropriate values of k (i.e., $k = 5 - 50$), the HDP/roleHDP can roughly converge to the best perplexity performance. However, for some extreme values of k (i.e., $k = 1, 100$), the HDP/roleHDP failed to converge. This issue was caused by some local optimum effects: from Figure 6.6 we can see that the converged log training likelihood tends to a maximum when the perplexity is minimized. When we initialized k to extreme values, it got stuck at the local optimum. Compared to the HDP/roleHDP, however, this local optimum effect is more severe for LDA. Therefore, the HDP/roleHDP demonstrated its better modelling ability—seen in the perplexity results in Figure 6.6—and is more robust to local optimum, by integrating over the topic cardinality.

6.7.2.2 The effect of role level.

In terms of perplexity, we can see that the roleHDP produced better results than the HDP. Moreover, the inclusion of role into the HDP provides some additional information. For example, we show in Figure 6.8 the four topic distributions specific to the four roles, and the top 3 example topics for each role from one roleHDP model. We can see the roleHDP reasonably captures the different topic distribution for each role. In this sense, the roleHDP is a promising model for the inclusion of role into the HDP framework.

6.7.2.3 The initial number of topics k.

Figure 6.7 shows that the HDP/roleHDP added topics for $k = 1$, and pruned topics for $k = 100$. Both initializations can potentially converge to the best value of K . Due to the local optima effect, however, it is better for us to begin with a larger number of topics than to begin from smaller number of topics, i.e., $k = 100$ normally has lower perplexity comparing to $k = 1$ in our results.

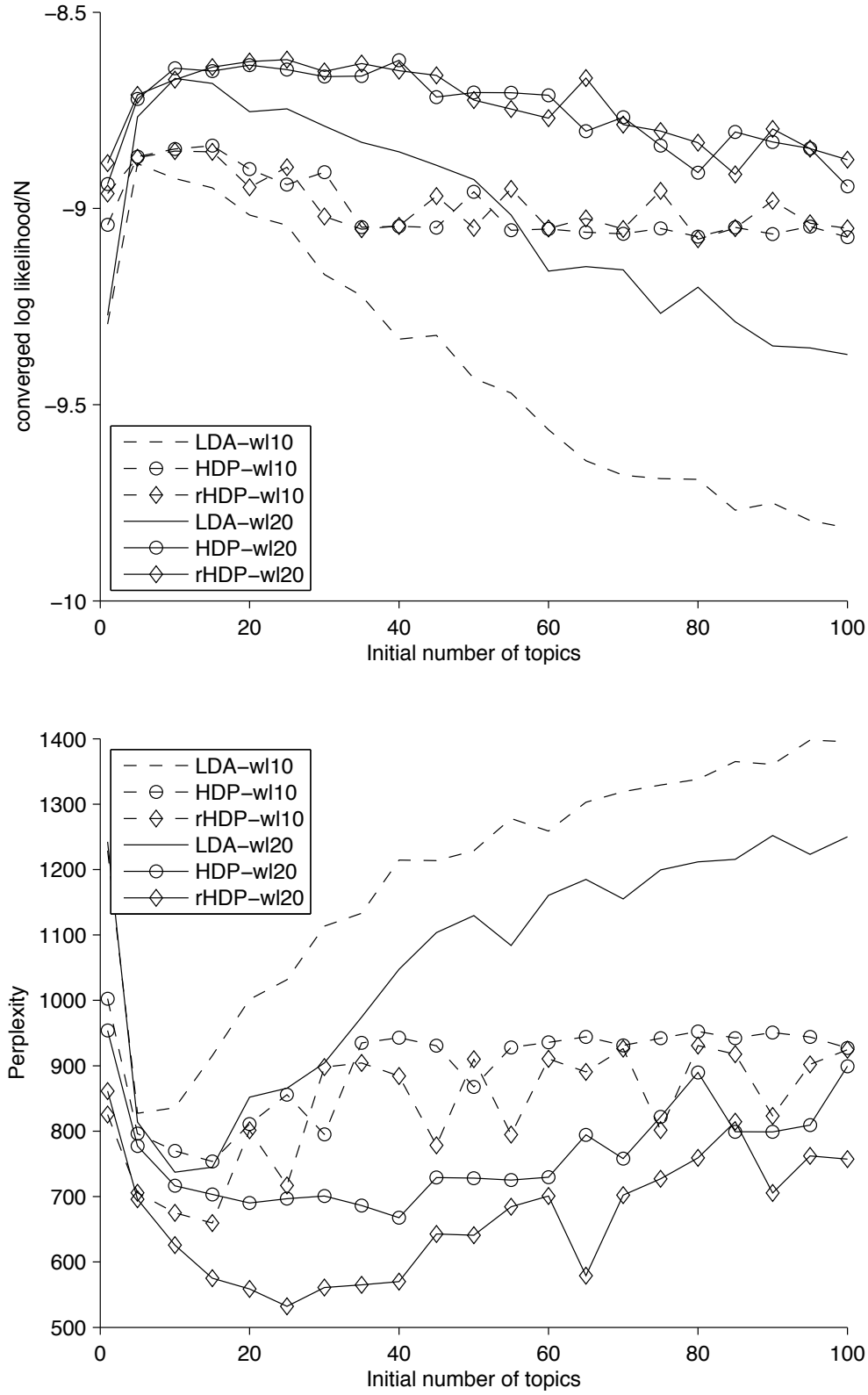


Figure 6.6: The empirical results of LDA and various HDP/roleHDP models using different parameters, where the x-axis is the k for $L = 10$ and $L = 20$; the y-axis is: (top) the converged train log likelihood per word on folds 2–4, (middle) the perplexity on fold 1, and (bottom) K. We here used $\beta = 1.0$ for all the results.

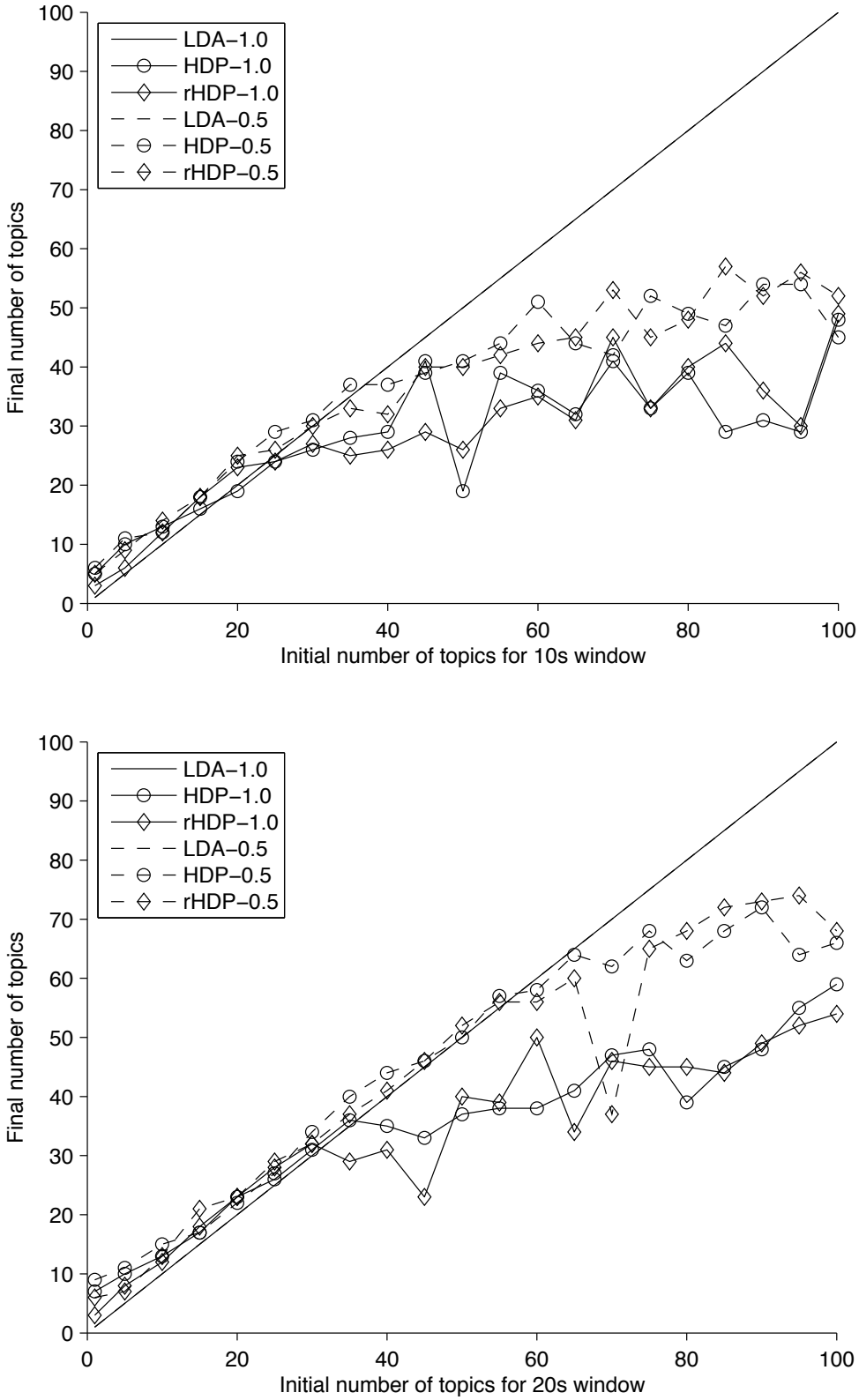
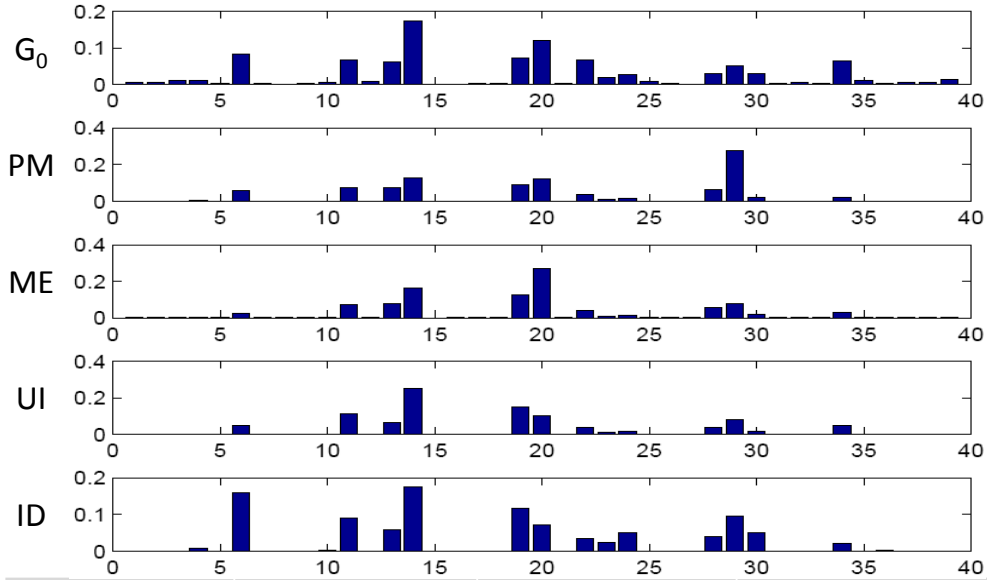


Figure 6.7: The empirical results of LDA and various HDP/roleHDP models using different parameters, where the x-axis is the k for $L = 10$ and $L = 20$; the y-axis is: (top) the converged train log likelihood per word on folds 2–4, (middle) the perplexity on fold 1, and (bottom) K.



PM		ME		UI		ID	
(29) 0.28	(14) 0.13	(20) 0.27	(14) 0.16	(14) 0.25	(19) 0.15	(14) 0.17	(6) 0.16
DESIGN	BUTTON	REMOTE	BUTTON	BUTTON	RECOGNITION	BUTTON	CHIP
MEETING	BUTTONS	CONTROL	BUTTONS	BUTTONS	SPEECH	BUTTONS	SIMPLE
PROJECT	CHANNEL	LOOK	CHANNEL	CHANNEL	VOICE	CHANNEL	ADVANCED
MINUTES	SCREEN	MARKET	SCREEN	SCREEN	L_C_D_	SCREEN	REGULAR
USER	SCROLL	CONTROLS	SCROLL	SCROLL	SCREEN	SCROLL	REMOTE
INTERFACE	VOLUME	MEAN	VOLUME	VOLUME	MICROPHONE	VOLUME	L_C_D_
PRESENT	MENU	EASY	MENU	MENU	CONTROLLER	MENU	BATTERY
PRODUCT	L_C_D_	PRODUCT	L_C_D_	L_C_D_	REMOTE	L_C_D_	SPEAKER
DESIGNER	REMOTE	DESIGN	REMOTE	REMOTE	PROBLEM	REMOTE	EXPENSIVE
LOOK	WHEEL	BUTTONS	WHEEL	WHEEL	TECHNOLOGY	WHEEL	INFRARED
START	MEAN	FANCY	MEAN	MEAN	COFFEE	MEAN	STATION
SURE	CHANNELS	IMPORTANT	CHANNELS	CHANNELS	SPEAKER	CHANNELS	SAMPLE
BIT	CONTROL	PERCENT	CONTROL	CONTROL	COST	CONTROL	BATTERIES
GUESS	PRESS	FIND	PRESS	PRESS	CONTROL	PRESS	COST
THANK	KIND	LOT	KIND	KIND	SYSTEM	KIND	CONTROL

Figure 6.8: Examples of topic distributions for different roles, and top 2 topics (shown as top 15 words) for each role. This is based on the roleHDP model with $k = 55, \beta = 0.5$, and $L = 10$. PM stands for project manager, ME for marketing expert, UI for user interface designer, and ID for industrial designer.

6.7.2.4 The prior parameter β.

The prior parameter β for the Dirichlet distribution plays an important role for the final value of K, with larger values of β leading to fewer final topics (see dash lines in Figure 6.7). Although for the HDP/roleHDP we do not need to manually set the number of topics K as in LDA, it is necessary to take care when initializing the value for β.

6.7.2.5 The document window length L.

The perplexity results for $L = 20$ are better, and more stable (with larger train likelihoods), than those for $L = 10$, with regard to different k. In addition, we found models with $L = 10$ suffered more severely from the local optima effect, for both LDA and HDP/roleHDP. This suggests the local optima effect may be partly caused by the length of document window we used here. The word co-occurrence in these relatively short documents are sparse, which makes it hard for the models to escape from a local optima.

We note, however, the two results for different window length may not be full comparable, because the length of both training and testing data are difference.

6.7.3 Topic Illustration and Dynamics

In the previous section, we did an empirical analysis on comparing LDA, the HDP and the roleHDP. We will further show some example topics extracted by LDA, the HDP, and the roleHDP. The parameters we used are $K = 25$, $\beta = 0.5$, $L = 20$, and $k = 25$ (the initial number of topics) for LDA, the HDP, and the roleHDP, which were trained using fold 2–4 of the AMI scenario meetings, with a fixed size vocabulary of 7,910 words. There are 26 topics in the final HDP model and 28 for the roleHDP. The perplexity on fold 1 of the AMI scenario meetings are 641.6, 582.2, and 433.5 for LDA, the HDP, and the roleHDP respectively.

Table D.1, Table D.2, and Table D.3 in the Appendix D show the example topics extracted by LDA, the HDP, and the roleHDP respectively. Given the domain specific AMI meeting corpus, it is not obvious which model produces more interpretable topics.

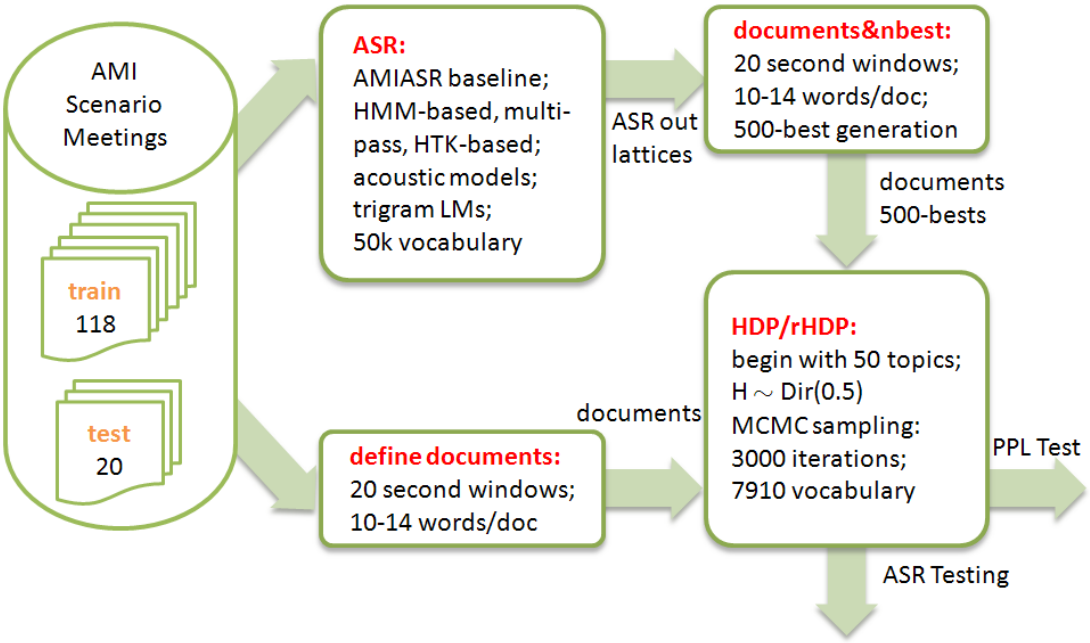


Figure 6.9: Experimental setup for evaluating the HDP and roleHDP in perplexity and ASR experiments.

6.7.4 Perplexity Experiments

In order to see the effect of the adapted LMs on perplexity, we trained three baseline LMs: the first one used the AMI n-gram training data, the second used the Fisher conversational telephone speech data (fisher-03-p1+p2), and the third used the Hub-4 broadcast news data (hub4-lm96). A fourth LM was trained using all three datasets. All the four LMs were trained with standard parameters using SRILM: trigrams, cut-off value of 2 for trigram counts, modified Kneser-Ney smoothing, interpolated model. A common vocabulary with 56,168 words was used for the four LMs, which has 568 out-of-vocabulary (OOV) words for the AMI test data.

The trained HDP and roleHDP models were used to adapt the above four baseline n-gram models respectively, using Equation (6.11) with $\mu = 0.5$. Different vocabularies were used by the HDP/roleHDP models compared with the baseline n-gram models. Only those words occurring in both the HDP/roleHDP vocabulary and the n-gram vocabulary were scaled using Equation (6.11). Table 6.2 shows the perplexity results for the adapted n-gram models. We can see both HDP- and roleHDP-adapted LMs produced significant reduction in perplexity, however there was no significant difference between using the HDP or roleHDP as the dynamic marginal in the adaptation.

Table 6.2: The perplexity results of HDP/roleHDP-adapted LMs.

LMs	Baseline	HDP-adapted	roleHDP-adapted
AMI	107.1	100.7	100.7
Fisher	228.3	176.5	176.4
Hub-4	316.4	248.9	248.8
AMI+Fisher+Hub-4	172.9	144.1	143.9

6.7.5 ASR Experiments

Finally, we investigated the effectiveness of the adapted LMs based on topic and role information from meetings on a practical large vocabulary ASR system. The AMI-ASR system [Hain et al., 2007] was used as the baseline system.

We began from the lattices for the whole AMI Meeting Corpus, generated by the AMI-ASR system using a trigram LM trained on a large set of data coming from Fisher, Hub4, Switchboard, webdata, and various meeting sources including AMI. We then generated 500-best lists from the lattices for each utterance. The reason why we used N-best rescoring instead of lattice rescoring is because the baseline lattices were generated using a trigram LM.

We adapted two LMs (Fisher, and AMI+Fisher+Hub4) trained in Section 6.7.4 according to the topic information extracted by HDP/roleHDP models based on the previous ASR outputs, using a moving document window with a length of 10 seconds. The adapted LM was destroyed after it was used to rescore the current N-best lists. Two adapted LMs together with the baseline LM were then used to rescore the N-best lists with a common language model weight of 14 (the same as for lattice generation) and no word insertion penalty.

Table 6.3 shows the WER results. LMs adapted by HDP/roleHDP both yield an absolute reduction of about 0.7% in WER. This reduction is significant using a matched-pair significance test [Jurafsky and Martin, 2009], as described in Section C, with $p < 10^{-15}$. However, again there was no significant difference between the HDP and the roleHDP.

To further investigate the power of HDP/roleHDP-adapted LMs, we trained a standard unigram, AMI-1g, on the AMI training data, which is the same data used for

Table 6.3: The %WER results of HDP/roleHDP-adapted LMs.

LMs	SUB	DEL	INS	WER
Fisher	22.7	11.4	5.8	39.9
role-FLM-adapted	22.5	11.1	5.9	39.5
AMI-1g-adapted	22.4	11.3	5.7	39.4
HDP-adapted	22.2	11.3	5.6	39.1
roleHDP-adapted	22.3	11.3	5.6	39.2
AMI+Fisher+Hub4	21.6	11.1	5.4	38.2
role-FLM-adapted	21.4	10.9	5.6	37.9
AMI-1g-adapted	21.3	11.0	5.4	37.8
HDP-adapted	21.2	11.1	5.3	37.6
roleHDP-adapted	21.2	11.1	5.3	37.5

HDP/roleHDP training. This unigram was trained using the same vocabulary of 7,910 words as that for HDP/roleHDP training. We then used this unigram as dynamic marginal to adapt the baseline LMs, also using the formula in Equation (6.11). The "AMI-1g-adapted" lines in Table 6.3 shows the WER results. We see, although AMI-1g-adapted LMs have lower WERs than that of the baseline LMs, HDP/roleHDP-adapted LMs still have better WER performances (with 0.2–0.3% absolute reduction) than AMI-1g-adapted. Significant testing indicates that both improvements for the HDP/roleHDP are significant, with $p < 10^{-6}$.

For the 5-fold cross validation setup experiments, We selected parameters with $k = 25$, $\beta = 1.0$, and $L = 20$ to train roleHDP models on each of the five folds of the AMI meeting data for the following ASR experiments. The topic information was extracted by the roleHDP models based on the previous ASR outputs, using a moving document window with the length of 20 seconds. We used Equation (6.11) to adapt the baseline LMs, with $\mu = 0.5$. Model interpolation (V1) and count interpolation (V2) were both used to deal with the vocabulary mismatch. The adapted LM was destroyed after it was used to rescore the current N-best lists. The rescoring used a common

language model weight of 14 (the same as for lattice generation).

Table 6.4 shows the WER results. We found consistent WER reductions in all the 5-fold ASR experiments on the AMI Meeting Corpus, using LMs adapted by the role-HDP. Although the absolute reductions are only about 0.2∼0.3% in WER, a significant testing using a matched-pair scheme [Jurafsky and Martin, 2009], as described in Section C, indicates that the reductions are all significant with $p < 0.01$. We also found that using count interpolation to deal with the vocabulary mismatch (V2) additionally provided a slightly better WER performance than the model interpolation version (V1).

ASR examples shown in Figure 6.10 illustrates the reasons for the improvements by adapting LMs based on the topic and role via the roleHDP. First, the meeting corpus we worked on is a domain-specific corpus with limited vocabulary, especially for those scenario meetings, with some words quite dominant during the meeting. So if we could roughly estimate the 'topic', and scale those dominant words correctly, then it is promising to improve the performance for LMs. Second, HDP/roleHDP models can reasonably extract topics, particularly on this domain-specific AMI Meeting Corpus. Third, the sentence-by-sentence style LM adaption further contributes to the improvements. Language models are dynamically adapted according to the changes of topics detected based on the previous recognized results. This can be intuitively understood as a situation where there are K unigram LMs, based on which we dynamically estimate one interpolated unigram LM to adapt the baseline LMs according to the context (topic). In this chapter, however, both the number of unigram models K and the unigram selected for one certain time are automatically determined by the roleHDP.

6.7.6 Experiments on the HDPLM

In this section, we describe our experiments and results for the HDPLM.

6.7.6.1 Data

We used the scenario part of the AMI Meeting Corpus. In total, there are 118 scenario meetings for training and 20 scenario meetings for testing. We applied a window of 30 seconds, as in role prediction experiments in Section 6.7.7, and we obtained a training data of 6,072 documents with 379,865 words and a testing data of 4,859 sentences with 104,682 words. Both the training and testing data contain the stop words in them. After reading the training data, the 6,072 documents are re-structured into a trie, and new documents are represented by words following a context h, resulting in 91,382

Table 6.4: The %WER results of roleHDP-adapted LMs, where V1 and V2 denote the model and count interpolations respectively for dealing with the vocabulary mismatch.

FOLD	LM	SUB	DEL	INS	WER
1	baseline	20.7	11.1	5.2	37.0
	roleHDP-V1-adapt	20.5	11.1	5.2	36.7
	roleHDP-V2-adapt	20.2	11.8	4.6	36.6
2	baseline	19.6	11.0	4.9	35.5
	roleHDP-V1-adapt	19.4	11.0	4.9	35.3
	roleHDP-V2-adapt	19.1	11.7	4.4	35.2
3	baseline	20.7	11.1	4.8	36.6
	roleHDP-V1-adapt	20.5	11.1	4.7	36.3
	roleHDP-V2-adapt	20.2	11.8	4.2	36.3
4	baseline	19.3	10.9	5.3	35.5
	roleHDP-V1-adapt	19.2	10.9	5.2	35.3
	roleHDP-V2-adapt	18.9	11.6	4.7	35.2
5	baseline	23.1	12.4	6.1	41.6
	roleHDP-V1-adapt	22.9	12.5	6.0	41.3
	roleHDP-V2-adapt	22.5	13.1	5.3	41.0
all	baseline	20.6	11.3	5.2	37.1
	roleHDP-V1-adapt	20.4	11.3	5.2	36.8
	roleHDP-V2-adapt	20.1	12.0	4.6	36.7

(A)	DOC	AGENDA TODAY DEFINE TALKED ENERGY KINETIC STUFF OPENING BATTERY COMPACT MINUTE PARTS MINUTES FUNCTIONAL DESIGN MEETING SIMPLE CHIP PRESENTATIONS DASH
	REF	***<s> OUR AGENDA WE'RE GOING TO DO AN OPENING I'M GOING TO REVIEW THE MINUTES OF </s>***
	BASE	<s> **RIGHT AND THEY WERE** GONNA DO THE OPENING A MINUTE OR IF YOU THE MINUTES OF </s>
	ADAPT	<s> **OUR AGENDA WE'RE** GONNA DO THE OPENING A MINUTE OR IF YOU THE MINUTES OF </s>
(B)	DOC	TEN TELETEXT BUTTONS NUMBERS BOSS AHEAD PAST PRETTY EASY AGES BUTTONS RECOGNITION FUNCTION REMOTE FINDING SCROLL CONTROL WHEEL REMOTE
	REF	***<s> YOU CAN INTRODUCE VOICE RECOGNITION BY UH FINDING BACK YOUR REMOTE </s>***
	BASE	<s> YOU CAN **AGES OF FOUR IS** RECOGNITION BY UH FINDING BACK YOUR REMOTE </s>
	ADAPT	<s> YOU CAN **AND USE THE VOICE** RECOGNITION BY UH FINDING BACK YOUR REMOTE </s>
(C)	DOC	ACTIVATE LIGHT LEADING CONSOLE STRONG BATTERY POWER KIND TECHNOLOGY EVALUATION FOCUSING CRITERIA L_ DUNNO C_D_ INSTRUCTIONS PANEL REQUIRES FEATURES HOUR L_C_D_ FORM LIGHT
	REF	***<s> WE COULD BECAUSE THE L. C. D. PANEL REQUIRES POWER AND THE L. C. D. IS A FORM OF A LIGHT </s>***
	BASE	<s> WE COULD BECAUSE THE L. C. D. PANEL REQUIRES **AN HOUR** AND THE L. C. D. IS A FORM OF LIGHT </s>
	ADAPT	<s> WE COULD BECAUSE THE L. C. D. PANEL REQUIRES **POWER** AND THE L. C. D. IS A FORM OF LIGHT </s>
(D)	DOC	NUMBERS CUSTOMISABLE CORNERS TABLE CHANNEL VOLUME FORTY VOLUME INTERESTED BRIGHTNESS CONTRAST READ SURE SHAPE REMOTE FRUIT BOWL IDEA STABLE CAT FEATURES
	REF	***<s> HAVE AN REMOTE IN THE SHAPE OF THE FRUIT OR A VEGETABLE OR WHAT EVER THEY LIKE </s>***
	BASE	<s> READ MORE IN THE SHAPE OF THE FRUIT BOWL OF **A STABLE** OR WHATEVER THE LIKE </s>
	ADAPT	<s> READ MORE IN THE SHAPE OF THE FRUIT BOWL OR **VEGETABLE** OR WHATEVER THEY LIKE </s>

(A)		(B)		(C)		(D)	
0.53	0.38	0.43	0.16	0.37	0.35	0.41	0.17
CHIP	MEETING	REMOTE	RECOGNITION	CHIP	REMOTE	BUTTON	FRUIT
BATTERY	DESIGN	CONTROL	SPEECH	BATTERY	SIGNAL	BUTTONS	BANANA
BATTERIES	MINUTES	LOOK	VOICE	BATTERIES	INFRARED	CHANNEL	SHAPE
ENERGY	PROJECT	FANCY	REMOTE	ENERGY	T_V_	SCREEN	COLOURS
SOLAR	USER	IMPORTANT	SAMPLE	SOLAR	BUTTON	SCROLL	SPONGY
KINETIC	INTERFACE	PERCENT	L_C_D_	KINETIC	CHIP	VOLUME	REMOTE
ADVANCED	PRESENTATION	USERS	SPEAKER	ADVANCED	BEEP	MENU	LOOK
SIMPLE	DESIGNER	CONTROLS	CONTROL	SIMPLE	LIGHT	L_C_D_	FEEL
STATION	THANK	EASY	SCREEN	STATION	CIRCUIT	WHEEL	VEGETABLES
REGULAR	START	TRUE	FIND	REGULAR	BOARD	PRESS	MEAN
L_C_D_	MARKETING	POINT	SENSOR	L_C_D_	INTERFACE	CHANNELS	VEGETABLE
POWER	INDUSTRIAL	FEEL	TECHNOLOGY	POWER	ACTUALLY	PUSH	FASHION
DOCKING	WORKING	BUTTONS	FEATURE	DOCKING	PRESS	FUNCTIONS	FRUITS
PRINT	THIRTY	INNOVATIVE	SIMPLE	PRINT	SEND	POWER	KIND
CELL	SURE	FIND	EASY	CELLS	USER	MEAN	CONTROL

Figure 6.10: Four ASR examples showing the roleHDP-adapted LM works better than the baseline LM. DOC is the document formed from the previous ASR output and used to extract topics, with the top 2 showing at the bottom accordingly, REF is the reference, and BASE and ADAPT are the ASR hypotheses of the baseline LM and roleHDP-adapted LM respectively.

new short documents totally. There are 7,532 words in the vocabulary.

6.7.6.2 Topic Illustration

For a comparison, we also use LDA for topic modelling. This corresponds to an extension from the bigram topic model in [Wallach, 2006] to an n-gram topic model, i.e., $n = 3$ in our experiments. We approximate the HDPLM to obtain this LDA trigram model by forcing the number of topics to be fixed during the Gibbs sampling. Both models use the same mechanism for inferring the hyperparameters.

We initialized LDA and the HDP with different initial number of topics, denoting different resultant models as LDAn and HDPn for LDA and the HDP respectively, with $n \in [1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200]$

Since we use more and shorter documents than usual for topic modelling, we are interested in seeing the final topics extracted by LDA and the HDP models. Table D.4 shows the examples of top 28 topics extracted by LDA30, and Table D.5 illustrates the examples of top 28 topics extracted by HDP30. It is hard, from Table D.4 and Table D.5, to tell which model extracts more reasonable topics. Comparing to extracted topics based on longer documents, i.e., those in Table D.1, Table D.2, and Table D.3, we find that longer documents typically lead to more interpretable topics than shorter documents in terms of topic interpretation, although this may not be true for other tasks such as word prediction in LMs.

6.7.6.3 Perplexity Results

We evaluate various n-gram LMs estimated from LDA and the HDP models in terms of perplexity on the testing data from the AMI Meeting Corpus. The testing data consists of 4,859 sentences with 104,682 words, and there are 1,130 out-of-vocabulary (OOV) tokens in the testing data.

Table 6.5 shows the perplexity results. We take the trigram HDLM as the baseline, approximated by limiting the number of topics in the HDPLM to be one during the inference using Gibbs sampling. This trigram HDLM can be regarded as the traditional trigram LM smoothed by additive smoothing using statistics from prior Dirichlet distribution. The trigram HDLM provides a perplexity of 373.8 on the testing data. For LDAn models, we see that further incorporating topic information into the HDLM reduces the perplexity, which decreases as the number of topics increases, bottoms at 146.8 for 30 topics, and then increase again as we increase the number of topics. For

HDPn models, the variance of perplexity is much smaller than that for LDAn models, which suggests the nonparametric modelling in the HDPn models is helpful for model selection.

We also show in Table 6.5 the perplexity results for state-of-the-art smoothed LMs, the IKNLM and the MKNLM. Without advanced smoothing techniques in the HDPLM, perplexity results of the HDPLM are worse the both the IKNLM (112.8) and the MKNLM (115.9).

Table 6.5: The perplexity results of the trigram HDPLM on the AMI Meeting Corpus.

LMs	**Perplexity**
trigram HDLM	373.8
trigram IKNLM	112.8
trigram MKNLM	115.9

LDA	**Perplexity**	**HDPLM**	**#Topics**	**Perplexity**
LDA1	373.8	HDP1	57	155.6
LDA10	163.0	HDP10	51	154.9
LDA20	149.0	HDP20	60	152.3
LDA30	146.8	HDP30	53	152.0
LDA40	148.7	HDP40	59	149.4
LDA50	151.4	HDP50	63	150.0
LDA60	154.9	HDP60	73	149.4
LDA70	158.9	HDP70	77	148.9
LDA80	163.0	HDP80	87	149.2
LDA90	167.4	HDP90	96	147.3
LDA100	172.1	HDP100	110	147.6
LDA200	213.8	HDP200	212	147.7

In this section, we carried our experiments on combining language models and topic models. Instead of still relying on topic inference during testing, we treat the

HDPLM as another smoothing technique for language models, and estimate a final smoothed LM in ARPA-format. We observed some reductions in perplexity on the AMI Meeting Corpus using the trigram HDPLM, comparing to the trigram HDLM.

6.7.7 Role Prediction using the roleHDP

We describe the experiments and results for role prediction using the roleHDP. The data and train/test split-up are the same as those in Section 6.7.1, except that we use here a different window length of 30 seconds. After removing those stop words in Section A, we obtain a training data set consisting of 6,072 documents with 96,361 words, and a testing data set of 1,107 documents with 19,676. There are 7,044 word types in the vocabulary. We initialized the roleHDP model with 50 topics, and $\beta = 0.5$ for Equation (2.38). After 3,000 iterations of Gibbs sampling, we finally have a roleHDP model with 11 topics.

To evaluate the performance of predicting roles by the roleHDP, we first train three n-gram LMs as the baseline models – two unigram LMs and one trigram LM – for each of the four roles, i.e., dividing the training data into four parts according to roles and training three n-gram LMs for each role. For unigram LMs, we use two different configurations: one includes the stop words in the training and testing data, while the other does not. For trigram LMs, we allow stop words during the training. During the testing, we predict the role for each testing document based on the predictive probabilities from LMs, according to Equation (6.18).

Additionally, we train four HDP topic models for the four roles, by initializing the HDP models with 50 topics and $\beta = 0.5$ for Equation (2.38). There are 3,000 iterations of Gibbs sampling to burn-in the HDP models. For each of the four roles, we estimate a final unigram LM from the HDP model according to Equation (6.20). This final unigram LM can be regarded as a mixture of unigram LMs derived from topics in the HDP model. Similarly, we use Equation (6.19) to select the most probable role for the testing document.

The roles were also manually predicted by an experienced human annotator, with extensive experience of working on the AMI corpus. We presented the annotator with readable test documents, including stop words[4].

Table 6.6 shows the results of role prediction on 1,107 testing documents. The human annotator correctly classified 527 out of 1107 testing documents, which cor-

[4]The testing site is available from `http://homepages.inf.ed.ac.uk/cgi/s0562315/predRoles.py`

responds to an accuracy result of about 47.6%. Although the roleHDP is indeed a unigram LM without stop words, the roleHDP-based prediction model yielded an accuracy of 41.7%, which is better than all the four baseline models. We also found that the inclusion of stop words in the unigram and trigram LMs helps. The HDP model, basically a mixture of unigram LMs, improves upon the traditional unigram LMs.

Table 6.6: The accuracy (%) of role prediction on the AMI Meeting Corpus using the roleHDP. The **STOPWORDS** column indicates whether or not we include stop words in the training and testing data.

MODEL	STOPWORDS	ACCURACY(%)
unigram	No	29.6
unigram	Yes	31.1
trigram	Yes	38.7
HDP-unigram	No	36.8
roleHDP	No	41.7
human	Yes	47.6

For each testing document, there are totally four possible predictive outputs. The accuracy results reported in Table 6.6 are evaluated using the best predictive output. For the roleHDP model, we also evaluate using the top-n outputs. They are 41.7%, 61.1%, 79.9%, and 100.0% for top one, two, three, and four respectively.

Figure 6.11 shows the curve of prediction accuracy for the roleHDP model, with regard to sampling iterations. Generally, the accuracy increases as the sampling goes on, and converges after reasonable number of iterations.

Each run of sample-based method will have slightly different results. Figure 6.12 shows the curves of another three runs of the roleHDP model. We also observe that incorporating the bigram role context information in Equation (6.23) (denoted by rbi) does not help in this task.

Table 6.7 and Table 6.8 show the confusion matrix of results for the roleHDP model and the human annotator, respectively. It is interesting to see that the human annotator did better on predicting documents for PM than the roleHDP model, while the roleHDP

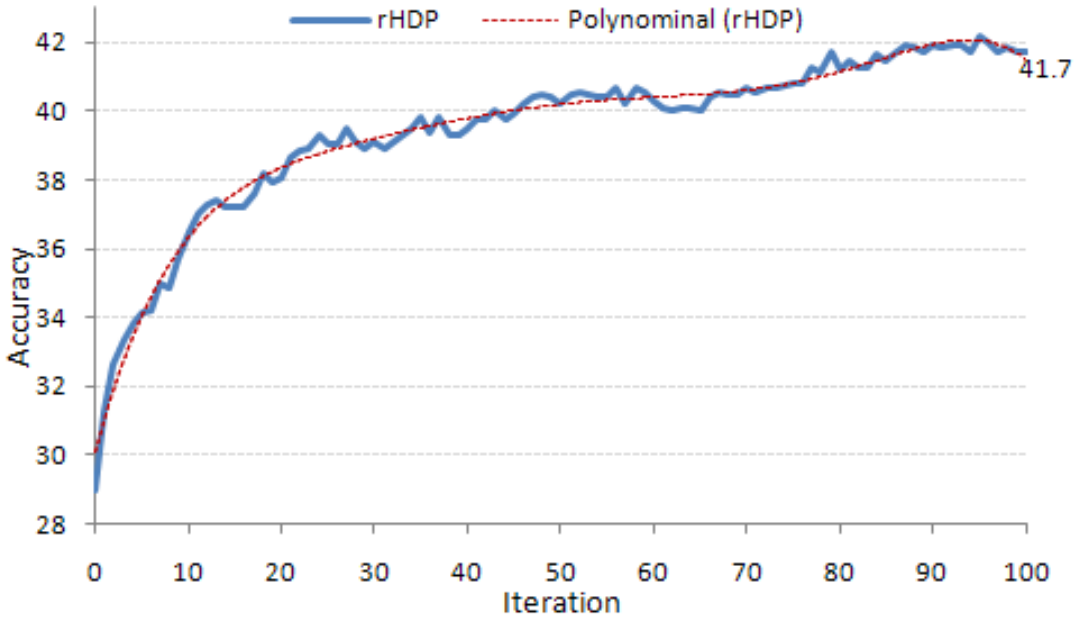


Figure 6.11: The prediction accuracy by the roleHDP changes over iterations (rHDP). We also show in the dotted line the polynomial regression of the accuracy curve.

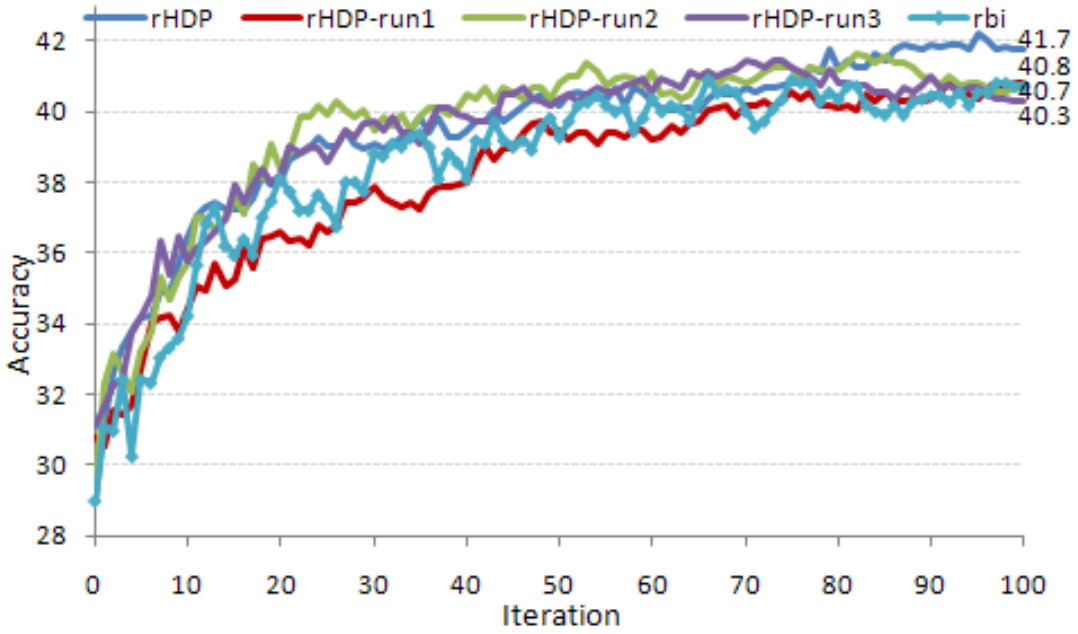


Figure 6.12: The prediction accuracy by the roleHDP changes over iterations. We show three runs of the roleHDP, rHDP-run1, rHDP-run2, and rHDP-run3 respectively, for role prediction, in addition to the best one (rHDP). We also plot the curve for roleHDP plus role bigram (rbi).

model did better on prediction documents for UI and ID than the human annotator.

Table 6.7: The confusion matrix of human result (41.7%). The rows correspond to the true roles, and the columns correspond to the predicted roles by the roleHDP.

	PM	**ME**	**UI**	**ID**	**NUM**	**ACC(%)**
PM	**146**	101	88	77	412	35.4
ME	55	**113**	77	38	283	39.9
UI	20	30	**101**	31	182	55.5
ID	21	40	67	**102**	230	44.3
SUM					**1107**	**41.7**

Table 6.8: The confusion matrix of human result (47.6%). The rows correspond to the true roles, and the columns correspond to the predicted roles by the roleHDP.

	PM	**ME**	**UI**	**ID**	**NUM**	**ACC(%)**
PM	**252**	49	63	48	412	61.2
ME	78	**110**	51	44	283	38.9
UI	37	28	**71**	46	182	39.0
ID	48	26	62	**94**	230	40.9
SUM					**1107**	**47.6**

6.8 Discussion and Summary

In this chapter, we successfully demonstrated the effectiveness of using the topic (and partly role) information to adapt LMs for ASR in meetings. The topics were automatically extracted using the nonparametric HDP model, which provides an efficient and flexible Bayesian framework for topic modelling. By defining the appropriate 'doc-

uments' for HDP models, we achieved a significant reduction in both perplexity and WER for a test set comprising about 11 hours of AMI meeting data.

We also investigated the use of shorter documents for topic modelling in this chapter. Does different document representations matter for topic models? According to our experiments, it is still not clear for this question, because it is not easy to evaluate the quality of extracted topics.

Sampling-based methods for the HDP and the roleHDP do not have deterministic results – each run may produce slightly different outputs. The variational methods, on the contrary, are believed to be more reliable in producing deterministic results.

To our understanding, the reasons for the significant improvements by adapted LMs based on the topic and role information via the HDP come from the following sources. First, the meeting corpus we worked on is a domain-specific corpus with limited vocabulary, especially for scenario meetings, with some words quite dominant during the meeting. So by roughly estimating the 'topic', and scaling those dominant words correctly, it is possible to improve LM accuracy. Second, HDP models can extract topics well, particularly on the domain-specific AMI Meeting Corpus. One interesting result we found is that different HDP/roleHDP models, though trained using various different parameters, did not result in significant differences in either perplexity or WER. By closely looking at the resulting topics, we found that some topics have very high probability regardless of the different training parameters. One characteristic of those topics is that the top words normally have very high frequency. Third, the sentence-by-sentence style LM adaption provides further improvements, to those obtained using the AMI-1g-adapted LMs in Table 6.3. Language models are dynamically adapted according to the changes of topics detected based on the previous recognized results. This can be intuitively understood as a situation where there are K unigram LMs, and we dynamically select one unigram to adapt the baseline LMs according to the context (topic). In this chapter, however, both the number of unigram models K and the unigram selected for a particular time are automatically determined by the HDP/roleHDP. Although this is unsupervised adaptation, it performs better than LM adaptation using static LMs trained on reference data.

One the other hand, the roleHDP had a similar accuracy to the HDP in terms of both perplexity and WER. Our interpretation for this is that we did not explicitly use the role information for adapting LMs, only using it as an additional DP level for sharing topics among different roles. As mentioned above, based on the AMI Meeting Corpus, which has a limited domain and consequently limited vocabulary words, this will not cause

much difference in the resulting topics, no matter whether HDP or roleHDP is used for topic modelling. Despite this, including the role information in the HDP framework can give us some additional information, such as the topics proportion specified to each role. This implies some scope to further incorporate role information into the hierarchical Bayesian framework for language modelling, for example by sampling the role randomly for each document, empirically analysing the differences between HDP and roleHDP, and explicitly using the role for language modelling. Another possibility for further investigation is about the prior parameter for Dirichlet distribution: can prior knowledge from language be used to set this parameter?

The conclusions we made in this chapter are as follows: 1) from the empirical analysis, we believe the HDP overall is a powerful and flexible framework for topic modelling, attributed by its nonparametric property and hierarchical structure; 2) the HDP is sensitive to the initialization of k, because of the local optima effect. The local optima effect is partly affected by the way we define a document; 3) we are convinced that the unsupervised LM adaptation framework using the HDP for meeting ASR, as presented here, is effective, at least on the AMI Meeting Corpus; 4) for ASR, a HDP/roleHDP model with lower empirical perplexity does not necessarily imply a lower WER. We observed WER results did not make much difference if we used a different HDP/roleHDP model for LM adaptation; 5) it is important to define an appropriate document for the HDP in topic-based LM adaptation for meeting ASR; 6) a combination of LM adaptation approaches seems promising; 7) although incorporating role information in the roleHDP did not produce better results than the HDP, we find it is useful for the roleHDP to predict roles in an artificial task; and 8) we observed some reductions in perplexity on the AMI Meeting Corpus using the trigram HDPLM, comparing to the trigram HDLM.

Chapter 7

Using Prosodic Features in Language Models for Meetings

Prosody has been actively studied as an important knowledge source for speech recognition and understanding. In this chapter, we are concerned with the question of exploiting prosody for language models to aid automatic speech recognition in the context of meetings. Using an automatic syllable detection algorithm, the syllable-based prosodic features are extracted to form the prosodic representation for each word. Two modelling approaches are then investigated. One is based on a factored language model, which directly uses the prosodic representation and treats it as a 'word'. Instead of direct association, the second approach provides a richer probabilistic structure within a hierarchical Bayesian framework by introducing an intermediate latent variable to represent similar prosodic patterns shared by groups of words. Four-fold cross-validation experiments on the ICSI Meeting Corpus show that exploiting prosody for language modelling can significantly reduce the perplexity, and also have marginal reductions in word error rate.

7.1 Introduction

Prosody, as discussed in Section 3.2.4, has long been studied as a knowledge source for speech understanding, and has been successfully used for a variety of tasks. This chapter is concerned with the question of exploiting prosody to aid automatic speech recognition (ASR) in the context of meetings. Speech in meetings is more natural and spontaneous than read or acted speech. The prosodic behaviours for speech in meetings are therefore much less regular. Three essential components in a state-of-

the-art ASR system, namely the acoustic model, language model (LM), and lexicon, can all potentially serve to accommodate prosodic features. In this chapter we are interested in exploiting prosodic features in language models for ASR in meetings.

The goal of a language model is to provide a predictive probability distribution for the next word conditioned on the strings seen so far, i.e., the immediately preceding $n-1$ words in a conventional n-gram model. In addition to the previous words, prosodic information associated with the audio stream, which is parallel to the word stream, can act as a complementary knowledge source for predicting words in LMs.

Due to the large vocabulary size in LMs (typically greater than 10,000 words), incorporating prosodic information in language models is more difficult than in other situations such as DA classification which has a much smaller number of target classes (typically several tens). To exploit prosody for LMs, a central question is how the relationship between prosodic features F and the word types W, $P(W|F)$, may be modelled. In this chapter, two models will be investigated, namely the factored language model (FLM) [Bilmes and Kirchhoff, 2003] and the hierarchical Bayesian model (HBM) [Gelman et al., 2004]. In the FLM-based approach, conditional probabilities $P(W|F)$ are directly estimated from the co-occurrences of words and prosody features via maximum likelihood estimation (MLE). The HBM-based approach provides a richer probabilistic structure by introducing an intermediate latent variable—in place of a direct association between words and prosodic features—to represent similar prosodic patterns shared by groups of words. This work is characterised by an automatic and unsupervised modelling of prosodic features for LMs in two senses. First, the prosodic features, which are syllable-based, are automatically extracted from audio. Second, the association of words and prosodic features is learned in an unsupervised way.

The rest of this chapter is organised as follows. The extraction of prosodic features is discussed in Section 7.2. Section 7.3 focuses on the modelling approaches, including the FLM-based method in Section 7.3.1 and the HBM-based method in Section 7.3.2. Experiments and results on the ICSI Meeting Corpus are reported in Section 7.4, followed by a discussion in the final section.

7.2 Prosodic Feature

A notable aspect of the prosodic features used here is that they are syllable-based. It is reasonable to address prosodic structures at the syllable level, because prosodic

features relating to the syllable reflect more clearly perceptions of accent, stress and prominence. Aylett has demonstrated the advantage of modelling prosody at the syllable level for dialogue act and hotspot categorization [Aylett, 2006], where a syllable finding algorithm based on band pass energy was used for syllable extraction. In this work, the syllable segments were automatically detected based solely on the parallel acoustic signals using an automatic syllable detection algorithm. The framework for the extraction of syllable-based prosodic features is shown in Figure 7.1, which follows an approach to automatic syllable detection suggested by Howitt [Howitt, 2000], which in turn was originated in work by Mermelstein [Mermelstein, 1975].

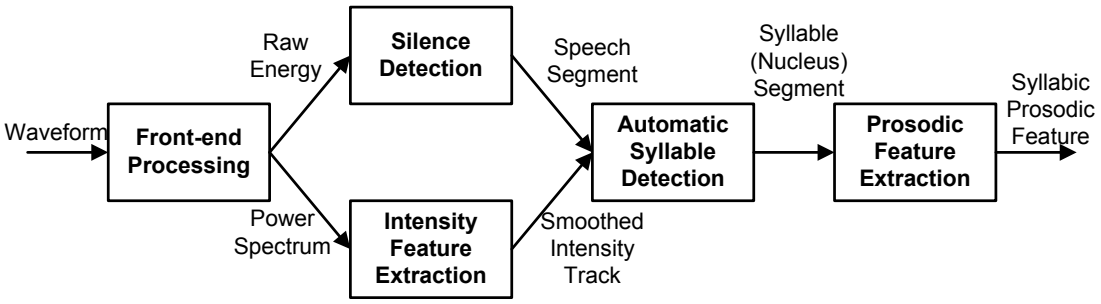


Figure 7.1: The framework of the extraction of syllable-based prosodic features.

1. **Front-end Processing** The speech signal was first framed using a 16 ms Hamming window with a shift period of 10 ms. The raw energy before windowing and pre-emphasis was computed for each frame and saved in log magnitude representation for subsequent silence detection. A 256-point FFT was used to compute the power spectrum.

2. **Silence Detection** The raw energy data was smoothed using a 6th-order low-pass 50 Hz filter. Each frame was classified into either speech or silence based solely on whether or not the log frame energy was above a threshold. A running window consisting of 10 consecutive frames was used to detect the onsets of speech and silence. The detected speech segments, which were further extended by 5 frames at both sides, were fed into the following syllable detection.

3. **Intensity Feature Extraction** A single measure of intensity was computed, following Howitt's adjusted features [Howitt, 2000]. A 300–900 Hz band-pass filter was used to filter out energy not belonging to vowels. By a weighted summation (converted to magnitude squared forms) of the spectral bins within 300–900 Hz frequencies from the spectrogram, an intensity track (converted back to decibels)

was computed for syllable detection, which again was smoothed by a low-pass 50 Hz filter to help reduce small peaks and noise.

4. **Automatic Syllable Detection** The recursive convex hull algorithm [Mermelstein, 1975], which is a straightforward and reliable syllable detection algorithm, was used to find the nuclei by detecting peaks and dips in the intensity track computed in the above step. The syllables were then obtained by extending the nuclei on both sides, until a silence or a boundary of adjacent nuclei is detected.

5. **Prosodic Feature Extraction** Four prosodic features were extracted for each syllable consisting of the duration of syllable, the average energy, the average F0, and the slope of F0 contour. F0 information was obtained using the ESPS `get_f0` program.

We ran vector quantization (VQ), with 16 codewords (labeled 's0' to 's15') over all the 892,911 observations of syllable-based prosodic features in the ICSI Meeting Corpus. Before running VQ, each feature was normalized to unit variance.

The syllables belonging to an individual word were obtained by aligning the word with the syllable stream according to a forced time alignment at the word level, and selecting those syllables whose centres were within the begin and end times of words. By concatenating relevant VQ indices for syllables, we obtained the symbolic representations of prosodic features at the word level, which can then serve as potential cues for language modelling. For example, the prosodic representation for word 'ACTUALLY' might be the symbol 's10s12s6', or 's10s15s6' in other contexts.

7.3 Modelling Approach

7.3.1 Factored Language Model

One straightforward method for modelling words and prosodic features is to use MLE based on the co-occurrences of words W and the prosodic representations F, i.e., training a unigram model $P(W|F) = \frac{\text{Count}(F,W)}{\text{Count}(F)}$. This unigram model can then be interpolated with conventional n-gram models. More generally, we can use the FLM [Bilmes and Kirchhoff, 2003] to model words and prosody deterministically. The FLM, initially developed to address the language modelling problems faced by morphologically rich or inflected languages, is a generalisation of standard n-gram language models, in which each word w_t is decomposed into a bundle of K word-related features (called

factors), $w_t \equiv f_t^{1:K} = \{f_t^1, f_t^1, \ldots, f_t^K\}$. Factors may include the word itself. Each word in an FLM is dependent not only on a single stream of its preceding words, but also on additional parallel streams of factors. Combining with interpolation or generalized parallel backoff (GPB) [Bilmes and Kirchhoff, 2003] strategies, multiple backoff paths may be used simultaneously. The FLM's factored representation can potentially accommodate the multimodal cues, in addition to words, for language modelling—in this case the prosodic representations. This configuration allows more efficient and robust probability estimation for those rarely observed word n-grams.

Supposing the word w_t itself is one of the factors $\{f_t^1, f_t^1, \ldots, f_t^K\}$, the joint probability distribution of a sequence of words $(w_1, w_2, \ldots, w_T)$ in FLMs can be represented as the formalism shown in Equation (7.1), according to the chain rule of probability and the n-gram-like approximation.

$$\begin{aligned} P(w_1, w_2, \ldots, w_T) &= P(f_1^{1:K}, f_2^{1:K}, \ldots, f_T^{1:K}) \\ &= \prod_{t=1}^{T} P(f_t^{1:K} | f_{t-1}^{1:K}, f_{t-2}^{1:K}, \ldots, f_1^{1:K}) \\ &\approx \prod_{t=1}^{T} P(w_t | f_{t-n+1:t-1}^{1:K}) \end{aligned} \tag{7.1}$$

There are two key steps to use FLMs. First an appropriate set of factor definitions must be chosen. We employed two factors: the word w_t itself and the syllable-based prosodic representation f_t, as shown in Figure 7.3(A). Second it is necessary to find

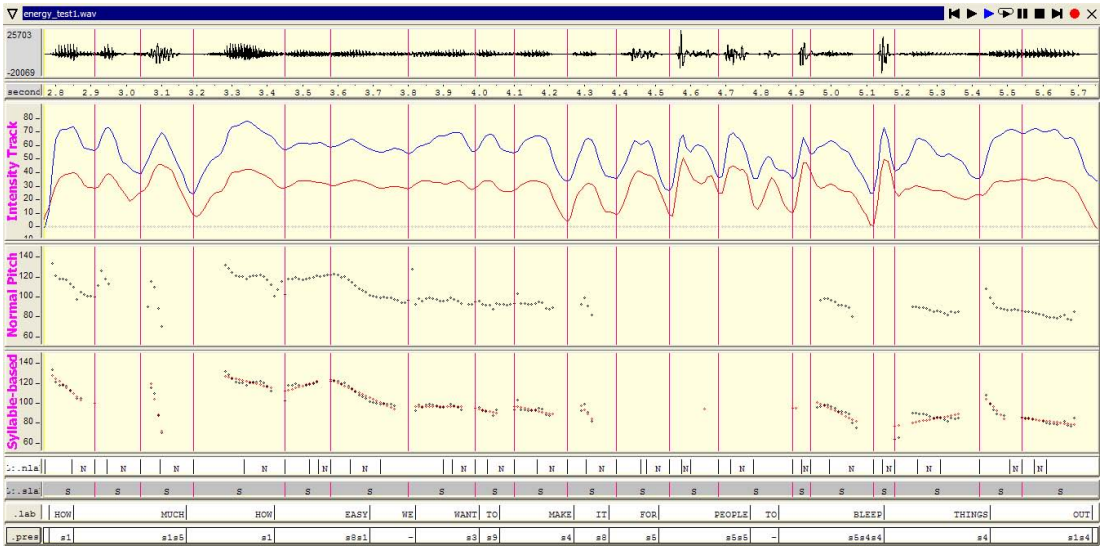


Figure 7.2: An example screenshot showing the system running to extract syllable-based prosodic features.

the suitable FLM models (with appropriate model parameters and interpolation/GPB strategy) over those factors. Although this task can be described as an instance of the structure learning problem in graphical models, we heuristically designed the model structure for FLMs. It is convenient to regard this FLM-based model as an interpolation of two conventional n-gram models $P(w_t|w_{t-1}, w_{t-2})$ and $P(w_t|w_{t-1}, f_t)$:

$$P_{\text{FLM}}(w_t|w_{t-1}, w_{t-2}, f_t) = \lambda_{\text{FLM}} P(w_t|w_{t-1}, w_{t-2}) + (1 - \lambda_{\text{FLM}}) P(w_t|w_{t-1}, f_t) \quad (7.2)$$

Figure 7.3(B) shows the parallel backoff graph used in the experiments for factors w_t and f_t. We perform the interpolation in a GPB framework, as depicted in Figure 7.3, manually forcing the backoff from $P(w_t|w_{t-1}, w_{t-2}, f_t)$ to two parallel paths by setting a very large value of *gtmin* for $P(w_t|w_{t-1}, w_{t-2}, f_t)$.

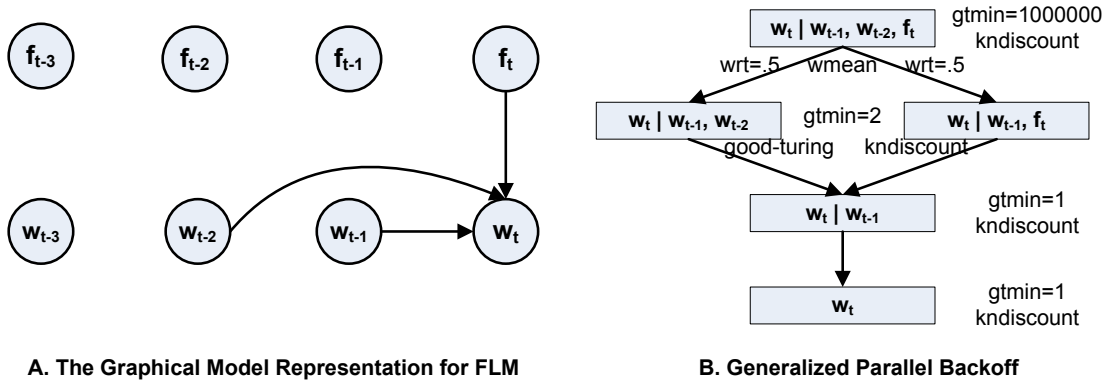


Figure 7.3: (A) A directed graphical model representation for the factor configuration in a FLM over factors including words w_t, the prosodic representations f_t. (B) The generalized parallel backoff graph for w_t and f_t used in the experiments.

7.3.2 Hierarchical Bayesian Model

We argue that it is essential but difficult to find intermediate symbolic representations to associate words and low-level prosodic features for language modelling. In this chapter, we have categorized syllable-based prosodic features into 16 classes, and represented the prosodic features for each word as a concatenation of indices for syllables belonging to that word. The FLM-based approach uses this prosodic information by directly associating word and prosodic representations. One limitation of this FLM-based approach is that there may be too many varieties of prosodic representations for individual words, due to the errors introduced by the automatic syllable detection and forced alignment. For example, the word 'ABSOLUTELY' in the ICSI Meeting Corpus has more than 100 different prosodic representations. Language models trained

via MLE using such prosodic representations will be more likely to overfit to the training data. Rather than the direct association of words and prosodic representations, we introduce a latent variable between word and prosody and assume a generative model that generates words from prosodic representations through the latent variable. This probabilistic generative models is investigated within the framework of hierarchical Bayesian models [Gelman et al., 2004].

Topic models have recently been proposed for document modelling to find the latent representation (*topic*) connecting documents and words. Latent Dirichlet allocation (LDA) [Blei et al., 2003] is one such topic model. As we discussed in Section 2.3.4.2 of Chapter 2, LDA is a three-level hierarchical Bayesian model, in which each document is represented as a random mixture over latent topics, and each topic in turn is represented as a mixture over words. The topic mixture weights θ are drawn from a prior Dirichlet distribution:

$$P(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \dots \theta_K^{\alpha_K - 1} \tag{7.3}$$

where $\alpha = \{\alpha_1, \dots, \alpha_K\}$ represents the prior observation count of the K latent topics with $\alpha_i > 0$. The LDA model is based on the "bag-of-words" assumption, that is, words in a document exchangeably co-occur with each other according to their coherent semantic meanings. In this sense, LDA can be considered as a probabilistic latent semantic analysis model. However what if we assume that words in a document exchangeably co-occur with each other according to their coherent prosodic patterns? This is the intuition of our use of LDA for the probabilistic association of words and prosody, which we call the prosody-topic model.

In a prosody-topic model, a document in the corpus is composed by including all those words that have the same prosodic representation (i.e., 's10s12s6'). The prosodic representation is then served as a common factor for that document. If we apply LDA over this corpus, we can extract the latent 'topics' connecting words and prosodic representations. Each topic is expected to have coherent prosodic patterns. Considering our prosodic representations in this chapter, for example, words in one individual topic are expected to have the same number of syllables whose pronunciations are similar. Unlike LDA, we need to explicitly retain the prosodic representations in the prosody-topic model. On the other hand, if we regard the prosodic representations as the 'authors' for corresponding documents, the prosody-topic model leads to the author-topic model [Rosen-Zvi et al., 2004], in which each document has only one unique author.

In short, the general idea of the prosody-topic model is that each prosodic repre-

sentation is represented by a multinomial distribution over latent topics, and each topic is represented as a multinomial distribution over words. Prosody thus serves the same role as semantics, being the guideline to cluster co-occurring words in a document. The goal of a prosody-topic model is to learn the distribution of words for each topic, which therefore finds the latent representations association the word and prosodic representations. The graphical model for prosody-topic model is shown in Figure 7.4, and the generative process for each document d can be described as follows.

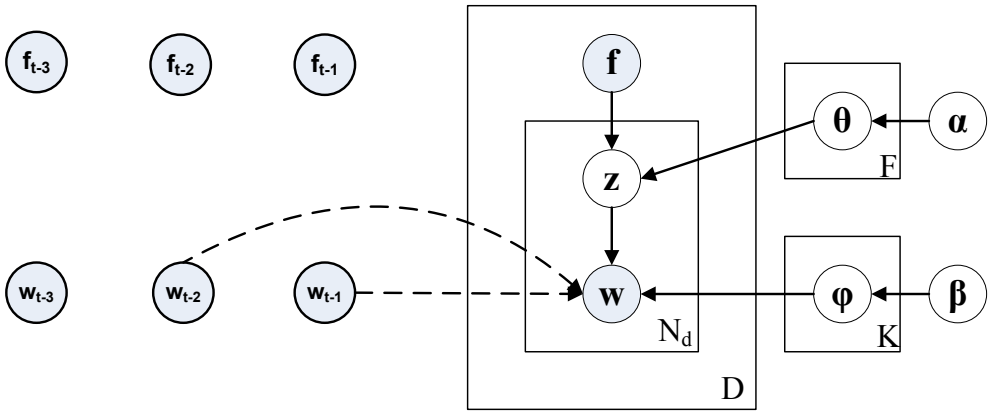


Figure 7.4: The graphical model representation for the prosody-topic model (right) and its interaction with n-gram model (left), where shaded nodes denote observed random variables, while unshaded ones denote latent variables or parameters. The boxes are 'plates' representing the replications of a corresponding substructure. The dashed arrow lines, representing the conditional dependence in the n-gram LM, are used to distinguish themselves from the solid arrow lines that represent the conditional dependence in the topic model.

1. Select the unique prosodic representation (author) label f for document d.
2. Choose topic proportions $\theta|\{f, \theta_{1:F}\}$ for document d according to f, each $\theta_f \sim Dirichlet(\alpha)$
3. For each of the N_d words w_n in document d:
   (a) Choose a topic $z_n|\theta \sim Mult(\theta)$.
   (b) Choose a word $w_n|\{z_n, \phi_{1:K}\} \sim Mult(\phi_{z_n})$, $\phi_{z_n} \sim Dirichlet(\beta)$.

Since each document only has a single author, the probability of words w_t given prosodic representations f_t in a prosody-topic model can be easily obtained by integrating out latent topics, as shown in Equation (7.4):

$$P_{\text{HBM}}(w_t|f_t) = \sum_{k=1}^{K} P(t_k|f_t)P(w_t|t_k) = \sum_{k=1}^{K} \theta_{f_t t_k}\phi_{t_k f_t} \tag{7.4}$$

where t_k is one of the K topics, $\theta_{f_t t_k}$ and $\phi_{t_k f_t}$ can be learned by approximate inference methods, such as variational EM or Gibbs sampling. This unigram-like probability can be interpolated with conventional n-gram models:

$$P_{\text{HBM}}(w_t|w_{t-1}, w_{t-2}, f_t) = \lambda_{\text{HBM}}P(w_t|w_{t-1}, w_{t-2}) + (1-\lambda_{\text{HBM}})P_{\text{HBM}}(w_t|f_t) \tag{7.5}$$

7.4 Experiments and Results

7.4.1 Meeting Corpus

The experiments reported here were performed using the ICSI Meeting Corpus [Janin et al., 2003], as described in Section 3.1.2 of Chapter 3. The ICSI Meeting Corpus has 75 naturally-occurring, unrestricted, and fairly unstructured research group meetings, each averaging about an hour in length. We performed our experiments using a four-fold cross-validation procedure in which we trained on 75% of the data and tested on the remaining 25%, rotating until all the data was tested on. The corpus was divided into four folds, first by ordering all the sentences in sequence, and then for each fold sequentially selecting every fourth sentence. After further removing the sentences that are too short in length to extract prosodic features, this procedure resulted in the data set summarised in Table 7.1.

7.4.2 Experimental Setup

We evaluated the FLM- and HBM-based approaches on the 4-fold cross-validation ICSI Meeting Corpus as described in Sect.7.4.1, in terms of perplexity (PPL) and word error rate (WER) respectively.

The FLM models were trained using the SRILM [Stolcke, 2002] toolkit[1], which has an extension for FLMs. Some modifications were made to the FLM toolkit regarding the manner of dealing with some special symbols such as '<s>', '</s>',

[1] http://www.speech.sri.com/projects/srilm/

Table 7.1: The summary of the four-fold cross-validation setup on the ICSI Meeting Corpus used in this chapter.

Fold	Number of Sentences	Number of Tokens
0-fold	27,985	209,766
1-fold	27,981	208,554
2-fold	27,968	208,294
3-fold	27,975	205,944

and 'NULL', e.g., we manually set $P(w_t|w_{t-1}, w_{t-2}, \text{NULL}) = P(w_t|w_{t-1}, w_{t-2})$, and scored the end-of-sentence '</s>' in perplexity calculations to account for the large number of short sentences in the meeting corpus. The FLM models share a common closed vocabulary of 50,000 word types with the AMI-ASR system [Hain et al., 2005]. The smoothing methods and parameters for FLM models are shown in Figure 7.3.

The prosody-topic models were trained using a publicly available Matlab topic modelling toolbox[2]. The algorithm for inference is Gibbs sampling [Griffiths and Steyvers, 2004], a Markov chain Monte Carlo algorithm to sample from the posterior distribution. We chose the number of topics $K = 100$, and ran the Gibbs sampling algorithm for 2500 iterations, which took around one hour to finish the inference on a 3-fold ICSI data. Instead of automatically estimating the hyperparameters α and β, we fixed these two parameters to be $50/K$ and 0.01 respectively, as in [Rosen-Zvi et al., 2004].

7.4.3 Perplexity and ASR Results

The PPL results were obtained by successively testing on the specific fold with the language model trained on the other three folds. The interpolation weights λ_{FLM} and λ_{HBM} were both set to 0.5. Table 7.2 shows the PPL results on the 4-fold cross-validation ICSI Meeting Corpus. Both FLM-based and HBM-based approaches produce some reduction in PPL, especially the HBM-based approach has over 10% relative reduction in PPL than the baseline trigram model. One interesting thing we found during analysing the PPL results sentence-by-sentence is that those having higher probabilities

[2]http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm

than baseline trigrams normally have reasonable prosodic representations for words, i.e., representing the right number of syllables in a word.

Table 7.2: PPL results for 4-fold cross-validation experiments. BASELINE-3G denotes the baseline trigram results using the FLM toolkit. FLM-3G-F denotes the results for the FLM-based model, while HBM-3G-F for the HBM-based prosody-topic model.

TRAIN-TEST	BASELINE-3G	FLM-3G-F	HBM-3G-F
123 – 0	78.4	73.6	70.5
023 – 1	78.9	73.9	70.7
013 – 2	78.3	73.4	70.1
012 – 3	78.3	73.3	70.8
AVERAGE	78.5	73.5	70.5

Table 7.3 shows the WER results of n-best rescoring on the ICSI Meeting Corpus. It should be noted that the BASELINE-2G WER results were obtained during the first-pass decoding of the AMI-ASR system using an interpolated bigram LM trained on seven text corpora including Hub4, Switchboard, ICSI Meeting, and a large volume (around 1GB in size) of web data. The lattices were generated using this interpolated bigram LM. By retaining the time information for candidate words, the lattices were then used to produce n-best lists with time stamps for subsequent rescoring experiments via the *lattice-tool* program in the SRILM toolkit. In our experiments, the 500-best lists were produced from the lattices, which were then aligned with the syllable streams to get prosodic representation for each word, and finally reordered according to scores of different interpolated LMs to search for the best hypothesis. Marginal reductions in WER were observed in our experiments.

7.5 Discussion and Summary

In this chapter we have investigated two unsupervised methods to exploit syllable-based prosodic features in language models for meetings. Experimental results on the ICSI Meeting Corpus showed our modelling approaches, both FLM-based and HBM-based, have significant reductions in PPL and marginal reductions in WER. The

Table 7.3: Word error rate results, which share the same notations as in Table 7.2, except that the BASELINE-2G column represents the baseline results from the first-pass AMI-ASR decoding using an interpolated bigram model.

TRAIN-TEST	BASELINE-2G	BASELINE-3G	FLM-3G-F	HBM-3G-F
123–0	29.8	29.5	29.2	29.1
023–1	29.6	29.3	29.1	29.0
013–2	29.5	29.2	29.0	28.9
012–3	29.4	29.2	29.1	29.0
AVERAGE	29.6	29.3	29.1	29.0

limited gains in WER may be partly caused by the following reasons. First, there are inevitably some errors in automatic syllable detection. It is hard for us to carry out evaluations on our syllable detection algorithm because of the lack of annotated data with syllable information. Second, additional errors are introduced by the forced alignment due to the overlapping cross-talk in meetings, which occasionally assigned an unreasonable number (i.e., more than 10) of syllables to a simple word. Third, the lattices were generated by an interpolated bigram model trained on a large corpus. This might prevent the recovery of more probable hypotheses from those n-best lists produced by a generalized LM, using specific LMs only trained on ICSI meeting data for rescoring.

Chapter 8

Conclusions and Future Directions

In this chapter, we summarize the main findings and conclusions of this thesis, and outline some future research directions.

8.1 Main Findings and Conclusions

This thesis has been concerned with the incorporation of useful multimodal cues from multiparty meetings with language models via hierarchical Bayesian models [Gelman et al., 2004], for an augmented language model for application to large vocabulary ASR of multiparty conversational speech from meetings. We carried out this study in the thesis from two aspects: the investigation on various hierarchical Bayesian models, and the investigation on various multimodal cues, for language modelling.

First, we summarize our investigation on various hierarchical Bayesian models for language modelling. In this thesis, we studied and compared the following statistical models, from both frequetist and Bayesian perspectives, to language modelling: interpolated Kneser-Ney smoothing LM (IKNLM) [Kneser and Ney, 1995], modified Kneser-Ney smoothing LM (MKNLM) [Chen and Goodman, 1999], power law discounting LM (PLDLM) in Chapter 5, latent Dirichlet allocation (LDA) [Blei et al., 2003] in Chapter 6, hierarchical Dirichlet process (HDP) [Teh et al., 2006] in Chapter 6, hierarchical Dirichlet LM (HDLM) [MacKay and Peto, 1994] in Chapter 4, hierarchical Dirichlet process LM (HDPLM) in Section 6.4, and hierarchical Pitman-Yor process LM (HPYLM) [Teh, 2006a] in Chapter 4. Figure 8.1 illustrates the relationship among the above mentioned statistical models. Under some conditions, one model can revert to another model, as indicated by the arrows in Figure 8.1.

Second, we summarize our investigation on various multimodal cues for language

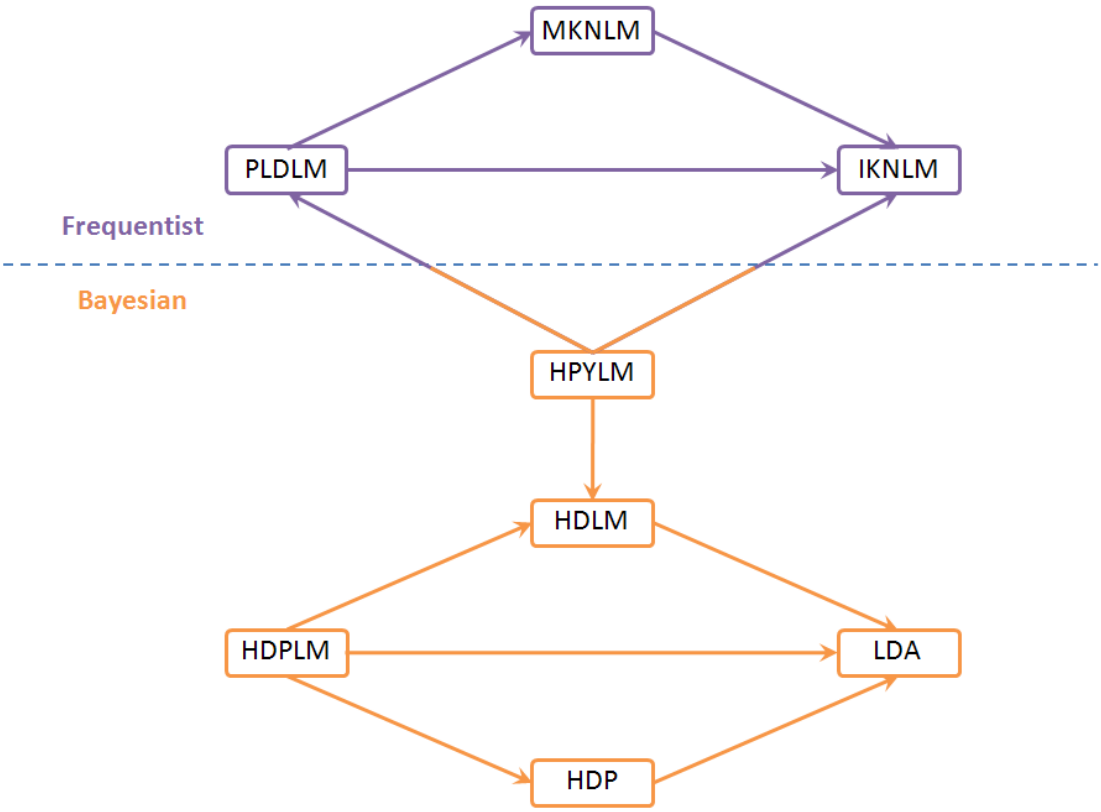


Figure 8.1: The relationship between various statistical models we investigated in this thesis, from both frequentist and Bayesian perspectives. Under some conditions, one model can revert to another model, as indicated by the arrows.

modelling. By using the various hierarchical Bayesian models, we studied in this thesis the following multimodal cues from multiparty meetings for language modelling: semantic context, or more specifically, topics, extracted by LDA/HDP models, in Chapter 6; participant role information in Chapter 6, and prosodic features in Chapter 7. In addition, we investigated how to estimate a better n-gram LM using lexical information in Chapter 4 and Chapter 5.

The main findings observed in this thesis are summarized as follows:

- Bayesian language models, such as the HPYLM, provide a Bayesian perspective for language model smoothing. The HPYLM has been proved to be helpful for ASR in Chapter 4, in terms of word error rate. One main reason for this is the power law characteristic provided by Pitman-Yor processes [Goldwater et al., 2006b; Teh, 2006b]. It is, however, computationally expensive to train and test a Bayesian language model. Therefore a parallel training algorithm, as described in Section 4.3, is normally required to apply the HPYLM on practical

ASR applications. The parallel training algorithm presented in Section 4.3 is a data parallelism scheme, and has been demonstrated to converge fast, as shown in Figure 4.3.

- An approximation to the HPYLM, called power law discounting, maintains the advantages of the HPYLM for language model smoothing. On the other hand, the PLDLM significantly reduces the computational complexity compared to the HPYLM. As shown in Section 5.3, the PLDLM preserves the important marginal constraints for language modelling. The PLDLM can be regarded as a generalisation of both the IKNLM and the MKNLM, by introducing more free parameters for discounting. The PLDLM directly uses a form of power law for discounting parameters. According to experimental results for the PLDLM in Section 5.4, we found that the same observation as that in [Chen and Goodman, 1999], that is, introducing more free parameters for smoothing can improve the performance for LM smoothing.

- Topic information can be useful for language modelling, as verified in Chapter 6. However, it is not trivial to incorporate the topic models (unigrams) with traditional n-gram LMs, especially for ASR applications. The widely used dynamic marginal adaptation [Kneser et al., 1997] requires to calculate a normalization term, which is time-consuming. Topic information is also required to be inferred during testing topic models, which again hinders the applications on ASR. Therefore, language models augmented by topic information are typically used in N-Best or lattice rescoring. Compared to LDA, the HDP has the ability to automatically determine the number of topics from training data. However, the HDP is sensitive to hyperparameters in determining the final number of topics, as demonstrated in Section 6.7.2. It is desirable to find a solution to more tightly combining topic models and language models. The HDPLM in Section 6.4 extends the model in [Wallach, 2006] to n-gram case using the HDP, and provides reductions in perplexity comparing to the HDLM.

- Social network information, such as participant role information in the AMI Meeting Corpus, is useful for topic modelling, according to the empirical analysis in Section 6.7.2. The roleHDP model in Section 6.5.2 has also been demonstrated to be useful in role prediction, as shown in Section 6.7.7. However, the inclusion of role in the HDP model did not help in practical ASR systems.

- Experiments in Chapter 7 show that using prosody for language modelling somewhat helps in reducing perplexity and WER. However, more sophistical models are required to handle these continuous prosodic features for language modelling. There are two problems: one is to extract and represent prosodic features, and the other is to incorporate these continuous prosodic features with LMs.

- Finally, we regard the central task we are fighting in this thesis as the *smoothing* for language models. we have tried various statistical models for better discounting, including PLDLM from frequentist's view, and HPYLM from Bayesian's view. We have also tried various multimodal cues to provide rich knowledge for better interpolation or adaptation for LMs. Overall, these various models and various multimodal cues are used to assign a non-zero probability to those unseen events, by either using better discounting techniques or making use of richer knowledge from multimodal cues.

To conclude, language models are an essential component of speech and language process systems, such as ASR and SMT. Smoothing plays an important role in language modelling. We in this thesis investigated various hierarchical Bayesian models to incorporate multimodal cues in meetings for language modelling. We observed the above interesting findings in this thesis during addressing the three questions we arose in Chapter 1.

8.2 Future Research Directions

There are some interesting research directions following this thesis that are worth further investigation in future, as outlined as follows:

- We have verified the effectiveness of the HPYLM for large vocabulary ASR tasks in terms of WER in Chapter 4. It will be interesting to see the generalisation of the HPYLM to SMT tasks in terms of BLEU [Papineni et al., 2002]. Log-linear based SMT systems exhibit a larger search space than ASR during the decoding. In this case, smoothing for language modelling in SMT may not be so important as in ASR.

- As shown in Equation (5.1), we derived the estimation of hyperparameters in the PLDLM, such as the discount parameter d, following the scheme for the IKNLM and the MKNLM. One direction to improve upon the PLDLM is to derive a new way to estimate hyperparameters, i.e., approaches derived from the HPYLM.

- We present the HDPLM in Chapter 6. It is attracting to tightly combine topic models with language models within a coherent hierarchical Bayesian framework. There are several issues to be taken into account and worth further investigations, for example, we could consider other advanced smoothing techniques in the HDPLM, such as absolute discounting; or we could more tightly combine topic models with Bayesian language models by placing both Dirichlet process and Pitman-Yor process as priors for each context.

- Regarding the use of prosody for language modelling, we could consider more tighter incorporation rather than simple interpolation, i.e., investigating the prosody-topic model in 7 and (Bayesian) language models in one united generative model within the hierarchical Bayesian framework. Moreover, meeting-specific cues will be taken into consideration for the prosody-topic model. For example, prosody encodes some information for dialogue acts (DA). DA in meetings normally has well-defined types. It is interesting to extend the prosody-topic model by investigating the relationship between word, prosody, and DA in one generative model.

Appendix A

Stop Word List

We show in Table A.1 the list of stop words we used to filter the vocabulary for topic modelling on the AMI Meeting Corpus in Chapter 6.

Table A.1: The list of stop words used for topic modelling on the AMI Meeting Corpus.

#.	’	’CAUSE	’KAY	’S	*	..	..#
..$	...	0	1	2	3	4	5
6	7	8	9	¡/s¿	¡s¿	A	ABOUT
ABOVE	ACCORDINGLY	ACROSS	AFTER	AFTERWARDS	AGAIN	AGAINST	AH
AL	AL.	ALL	ALLOWS	ALMOST	ALONE	ALONG	ALREADY
ALRIGHT	ALSO	ALTHOUGH	ALWAYS	AM	AMONG	AMONGST	AN
AND	ANOTHER	ANY	ANYBODY	ANYBODY’S	ANYHOW	ANYONE	ANYTHING
ANYWAY	ANYWHERE	APART	APPEAR	APPROPRIATE	ARE	AREN’T	AROUND
AS	ASIDE	ASSOCIATED	AT	AVAILABLE	AWAY	AWFULLY	A_
B	BACK	BE	BECAME	BECAUSE	BECOME	BECOMES	BECOMING
BEEN	BEFORE	BEFOREHAND	BEHIND	BEING	BELOW	BESIDE	BESIDES
BEST	BETTER	BETWEEN	BEYOND	BOTH	BRIEF	BUT	BY
BYE	B_	C	CAME	CAN	CAN’T	CANNOT	CANT
CAUSE	CAUSES	CERTAIN	CHANGES	CO	COME	CONSEQUENTLY	CONTAIN
CONTAINING	CONTAINS	CORRESPONDING	COULD	COULDN’T	CURRENTLY	C_	D
DAY	DESCRIBED	DID	DIDN’T	DIFFERENT	DO	DOES	DOESN’T
DOING	DON’T	DONE	DOWN	DOWNWARDS	DURING	D_	E
E.G	EACH	EG	EH	EIGHT	EITHER	ELSE	ELSEWHERE
ENOUGH	EQ	ET	ETC	EVEN	EVER	EVERY	EVERYBODY
EVERYONE	EVERYTHING	EVERYWHERE	EX	EXAMPLE	EXCEPT	E_	F
FAR	FEW	FIFTH	FIRST	FIVE	FOLLOWED	FOLLOWING	FOR
FORMER	FORMERLY	FORTH	FOUR	FROM	FURTHER	FURTHERMORE	F_
G	GET	GETS	GIVE	GIVEN	GIVES	GO	GOES
GOING	GONE	GONNA	GOOD	GOT	GREAT	H	HAD
HADN’T	HARDLY	HAS	HASN’T	HAVE	HAVEN’T	HAVING	HE
HE’S	HELLO	HENCE	HER	HERE	HERE’S	HEREAFTER	HEREBY
HEREIN	HEREUPON	HERS	HERSELF	HEY	HI	HIM	HIMSELF
HIS	HITHER	HM	HM-HMM	HMM	HOW	HOWBEIT	HOWEVER
HUH	I	I’D	I’LL	I’M	I’VE	I.E	IE
IF	IGNORED	IMMEDIATE	IN	INASMUCH	INC	INDEED	INDICATE
INDICATED	INDICATES	INNER	INSOFAR	INSTEAD	INTO	INWARD	IS
ISN’T	IT	IT’D	IT’LL	IT’S	ITS	ITSELF	I_
J	JUST	J_	K	KEEP	KEEPING	KEPT	KNEW
KNOW	KNOWS	K_	L	LAST	LATTER	LATTERLY	LEAST
LESS	LEST	LET	LET’S	LIFE	LIKE	LITTLE	LONG
LTD	M	MADE	MAKE	MAKING	MAN	MANY	MAY
MAYBE	ME	MEANWHILE	MEN	MIGHT	MM	MM-HM	MM-HMM
MORE	MOREOVER	MOST	MOSTLY	MR	MUCH	MUST	MY
MYSELF	M_	N	NAME	NAMELY	NEAR	NECESSARY	NEED
NEITHER	NEVER	NEVERTHELESS	NEW	NEXT	NINE	NO	NO-ONE
NOBODY	NONE	NOONE	NOR	NORMALLY	NOT	NOTHING	NOVEL
NOW	NOWHERE	N_	O	OF	OFF	OFTEN	OH
OH,BUT	OKAY	OLD	ON	ONCE	ONE	ONE’S	ONES
ONLY	ONTO	OOH	OOPS	OR	OTHER	OTHERS	OTHERWISE
OUGHT	OUR	OURS	OURSELVES	OUT	OUTSIDE	OVER	OVERALL
OWN	O_	P	PARTICULAR	PARTICULARLY	PEOPLE	PER	PERHAPS
PLACED	PLEASE	PLUS	POSSIBLE	PROBABLY	PROVIDES	PUT	Q
QUE	QUITE	R	RATHER	REALLY	RELATIVELY	RESPECTIVELY	RIGHT
R_	S	SAID	SAME	SAW	SAY	SAYS	SECOND
SECONDLY	SEE	SEEM	SEEMED	SEEMING	SEEMS	SELF	SELVES
SENSIBLE	SENT	SERIOUS	SEVEN	SEVERAL	SHALL	SHE	SHE’D
SHE’S	SHOULD	SHOULD’VE	SHOULDN’T	SINCE	SIX	SO	SOME
SOMEBODY	SOMEHOW	SOMEONE	SOMETHING	SOMETHING’S	SOMETIME	SOMETIMES	SOMEWHAT
SOMEWHERE	SORRY	SPECIFIED	SPECIFY	SPECIFYING	STATE	STILL	SUB
SUCH	SUP	T	TAKE	TAKEN	TAKES	THAN	THAT
THAT’D	THAT’LL	THAT’S	THE	THEIR	THEIRS	THEM	THEMSELVES
THEN	THENCE	THERE	THERE’D	THERE’LL	THERE’RE	THERE’S	THEREAFTER
THEREBY	THEREFORE	THEREIN	THEREUPON	THESE	THEY	THEY’D	THEY’LL
THEY’RE	THEY’VE	THING	THING’S	THINGS	THINK	THIRD	THIS
THOROUGH	THOROUGHLY	THOSE	THOUGH	THREE	THROUGH	THROUGHOUT	THRU
THUS	TIME	TO	TOGETHER	TOO	TOOK	TOWARD	TOWARDS
TRY	TWICE	TWO	T_	U	UH	UH-HUH	UH-OH
UH-UH	UM	UM-HMM	UNDER	UNLESS	UNTIL	UNTO	UP
UPON	US	USE	USED	USEFUL	USES	USING	USUALLY
V	VALUE	VARIOUS	VERY	VIA	VIZ	VS	V_
W	WANNA	WANT	WAS	WASN’T	WAY	WE	WE’D
WE’LL	WE’RE	WE’VE	WELL	WENT	WERE	WEREN’T	WHAT
WHAT’S	WHATEVER	WHEN	WHENCE	WHENEVER	WHERE	WHERE’S	WHEREAFTER
WHEREAS	WHEREBY	WHEREIN	WHEREUPON	WHEREVER	WHETHER	WHICH	WHILE
WHITHER	WHO	WHO’S	WHOEVER	WHOLE	WHOM	WHOSE	WHY
WILL	WITH	WITHIN	WITHOUT	WOAH	WON’T	WORK	WORLD
WOULD	WOULDN’T	WOW	X	Y	YEAH	YEAH.	YEAR
YEARS	YEP	YES	YET	YOU	YOU’D	YOU’LL	YOU’RE
YOU’VE	YOUR	YOURS	YOURSELF	YOURSELVES	Z	ZERO	Z_

Appendix B

The 5-fold Cross Validation Split-up

We include in the Table B.1 the meeting identities from the scenario AMI Meeting Corpus for the 5-fold cross validation experiments in Chapter 4 and Chapter 6.

Table B.1: Meeting IDs for the 5 folds of the AMI scenario meetings. The initial letter 'S' represents the scenario meetings.

FOLD	**No.**	**MEETINGS**
1	32	S2004[A-D] S2005[A-D] S2007[A-D] S2013[A-D] S2014[A-D] S3007[A-D] S3008[A-D] S3009[A-D]
2	28	S2006[A-D] S2009[A-D] S2011[A-D] S2016[A-D] S3005[A-D] S3010[A-D] S3011[A-D]
3	24	S1009[A-D] S2003[A-D] S2012[A-D] S2015[A-D] S3006[A-D] S3012[A-D]
4	28	S1007[A-D] S1008[A-D] S2002[A-D] S2008[A-D] S2010[A-D] S3003[A-D] S3004[A-D]
5	25	S1000[A-D] S1001[A-D] S1002[B-D] S1003A S1003C S1003D S1004[A-D] S1005[A-C] S1006[A-D]

Appendix C

Matched-Pair Significance Test

We use in this thesis the matched-pair significance test method [Jurafsky and Martin, 2009] for significance testing of WER results in the ASR experiments.

Suppose baseline and newsystem are the filtered hypothese files, generated by the NIST scoring tool sclite. We use the script compare-sclite from the SRILM toolkit [Stolcke, 2002] to compute the matched pairs:

```
compare-sclite -r refs -h1 baseline -h2 newsystem
```

Let M be the number of times the two systems' hypotheses differ, and N be the number of times new system improves upon baseline. We compute the probability that a fair coin would have come out heads at least N times in totally M trials: $P(k \geq N|M)$ as the significance p-value, with $p < 0.01$ to be significant, and $p < 0.05$ to be significant but weak.

Appendix D

Example Topics

In this appendix, we show some example topics extracted by those topic models that we have investigated in Chapter 6.

Table D.1: Example topics by LDA on fold 2–4 of the AMI Meeting Corpus.

PHONE	RECOGNITION	TRIPLE	DRAW	RUBBER	COMMAND	DESIGN	POINT	TWELVE	PRODUCT	PINEAPPLE	HAND	GREEN
LOOK	SPEECH	REAL	PEN	PLASTIC	INTERFACE	MEETING	TRUE	TWENTY	MARKET	REMOTE	CURVED	COBRA
SCHOOL	REMOTE	REMOTE	ANIMAL	CASE	SCREEN	USER	SCALE	EUROS	EVALUATION	SIZE	MEAN	LOT
FASHIONABLE	L_C_D_	BANANA	CAT	WOOD	CUT	INTERFACE	SPEAKER	FIFTY	COST	BIGGER	SORT	FEEL
MEAN	CONTROL	CONTROL	BIT	TITANIUM	T_F_T_	PROJECT	SAMPLE	POINT	DESIGN	MUSHROOM	KIND	HIGH
IPODS	VOICE	REACTION	FAVOURITE	SPECIAL	COMPONENT	REMOTE	WAIT	COST	PROJECT	PITCH	BIT	KIND
STRAWBERRY	MEAN	MANDO	BOARD	MATERIAL	CARD	MINUTES	CAT	PRICE	LOOK	CONTROL	SHAPE	MICROPHONE
HIGH	SCREEN	SOUNDS	DRAWING	COLOUR	DECIDED	MARKETING	FANCY	EURO	REMOTE	BASE	BUTTONS	TECHNOLOGICALLY
T_V_	T_V_	SPANISH	DOG	BUTTONS	BIG	DESIGNER	CATS	REMOTE	MONEY	KIND	SINGLE	CHRISTA
LOOKS	LOSE	COOL	PRETTY	SPONGY	LOST	CONTROL	POINTS	SELL	MEAN	SMALLER	CURVE	ARRAY
END	TECHNOLOGY	SIZE	DIGITAL	CURVED	USER	PRESENTATION	FALSE	MILLION	MARKETING	ERASE	SIDE	PUMPKIN
CHARGER	HARD	MEAN	LOOKS	REMOTE	SHINE	WORKING	NICE	HALF	PERCENT	SMALL	REMOTE	EASY
REMOTE	KIND	R_C_	WRITE	MEAN	REMEMBER	START	LOOK	PRODUCTION	SURE	MEAN	ROUND	NATURAL
FIND	LOOK	SURE	DUNNO	KIND	PROBLEM	PRODUCT	MICROPHONE	PROFIT	QUESTION	CUTE	FLAT	PEPPER
LOST	COST	DASH	GUESS	COLOURS	POWERPOINT	LOOK	MAXIMUM	CONTROL	CRITERIA	CONTROLS	HOLD	BIG
TECH	EXPENSIVE	HMMM	EASY	HARD	SIZE	INDUSTRIAL	REMOTE	SIXTEEN	FINANCE	BRILLIANT	IDEA	VOICE
SHEEP	BUTTON	REACTOR	WHITEBOARD	CONTROL	CALLED	FUNCTIONAL	ACTUALLY	MARKET	PRETTY	TREND	ACTUALLY	DEFINITELY
FORM	DISPLAY	FUTURE	NICE	DOUBLE	LINE	TECHNICAL	LOVE	BUDGET	SELL	DIRECTION	BIG	CENTIMETRES
CONTENT	SURE	WEIGHT	SURE	DESIGN	PICTURE	SURE	BAD	SELLING	PRESENTATION	READ	DESIGN	WANTED
BUY	DESIGN	DOUBLE	SLOW	MATERIALS	CONCEPT	GUESS	ALLERGIC	FIFTEEN	PRICE	VOMIT	LEFT	ELEPHANTS
BLUE	BATTERY	BUTTON	SCROLL	MEETING	BREAK	COLOUR	REMOTE	FRUIT	MINUS	REMOTE	CHIP	
TONGUE	BATTERIES	BUTTONS	BUTTON	MINUTES	STARTED	COLOURS	CONTROL	REMOTE	HOMER	CONTROL	REGULAR	
EATING	SOLAR	SCREEN	CHANNEL	PROJECT	STEM	YELLOW	FORTY	SPONGY	TELETEXT	BUTTONS	ADVANCED	
GIRAFFE	ENERGY	MENU	WHEEL	THANK	CHOICE	BLACK	RECOGNITION	BANANA	IRRITATING	T_V_	SIGNAL	
BLUETOOTH	REMOTE	REMOTE	BUTTONS	EMAIL	MARTIAN	BUTTONS	MARKET	LOOK	INTERNET	CONTROLS	REMOTE	
LEAVES	KINETIC	CHANNEL	CHANNELS	INFORMATION	COOL	REMOTE	GROUP	KIND	SAN	USER	INFRARED	
TREE	STATION	VOLUME	VOLUME	FOLDER	TEAMWORK	SPECIAL	TWENTY	MEAN	COMPUTER	EASY	BUTTON	
BEAGLE	POWER	L_C_D_	PUSH	GUYS	PENGUIN	RED	YOUNGER	SHAPE	DIEGO	MEAN	CIRCUIT	
INTERESTING	CELLS	PRESS	L_C_D_	SURE	KIND	KIND	MEAN	FASHION	ACTUALLY	TELEVISION	SPEAKER	
SPONGY	CONTROL	CONTROL	MENU	BIT	TOGGLE	MEAN	L_C_D_	IDEA	LOGO	LOT	BEEP	
MODEL	RECHARGEABLE	FUNCTIONS	TURN	ROOM	COMPUTER	LOGO	SPEECH	CONTROL	NINJA	CHANNEL	SIMPLE	
SORT	DYNAMO	MEAN	MEAN	PRESENTATION	SCROLL	CONTROL	PERCENT	FEEL	CLICKING	FIND	SAMPLE	
DESIGN	IDEA	T_V_	SWITCH	LOT	DEFINITELY	GREEN	TARGET	COLOURS	NICE	DESIGN	PRINT	
ANIMAL	MEAN	TOP	DISPLAY	ACTUALLY	AFRAID	BLUE	AGE	FANCY	WORRY	KIND	BUTTONS	
FLUFFY	BUY	OPTIONS	SIDE	GUESS	HMMM	GREY	LOOK	VEGETABLES	MISS	BUTTON	LIGHT	
PROMOTING	HAND	EASY	KIND	COOL	REAL	LIGHT	CONTROLLER	FRUITS	GUYS	LOOK	BATTERY	
DEVICES	LIGHT	POWER	PRESS	FIND	REACTION	BUTTON	SCREEN	VEGETABLE	CALIFORNIA	VOLUME	T_V_	
JESSY	SURE	SETTINGS	NUMBERS	KIND	YELLOW	LOOK	T_V_	DUNNO	NEEDED	ACTUALLY	L_C_D_	
ACTUALLY	DOCKING	KIND	INTEGRATED	THANKS	CHOOSE	RUBBER	KIND	IMPORTANT	TECHNOLOGY	FUNCTIONS	EXPENSIVE	
FEELING	SOURCE	TELETEXT	TELETEXT	SAVE	REMOTE	COMPANY	YOUNG	MOBILE	BUCKS	CHANGE	BOARD	

Table D.2: Example topics by the HDP on fold 2–4 of the AMI Meeting Corpus.

INJURY	BATTERY	WATCHES	SCREEN	BEEP	REMOTE	SCHOOL	BRILLIANT	BUTTON	REMOTE	MUSHROOM	BRUSSELS	REAL
STRAIN	BATTERIES	BLUE	CUT	CALL	TWENTY	LOOK	SNAIL	BUTTONS	CONTROL	KIND	AFRO	REACTION
USERS	ENERGY	NATURAL	STUDYING	DEEP	EUROS	ELEMENTARY	SHELL	CHANNEL	T_V_	PINEAPPLE	BEAT	NICE
AFFECTED	SOLAR	PHONE	WORDS	BLACKS	TWELVE	MACS	SIT	SCREEN	TELEVISION	FRUIT	HMMM	SAN
QUARTER	KINETIC	SMALLER	T_F_T_	SPEAKER	FIFTY	IPODS	READ	VOLUME	BUTTON	SPORTS	ORCHESTRA	MISS
REPEATED	POWER	TONGUE	PSYCHOLOGY	SILVERS	MARKET	FASHIONED	VOMIT	MENU	BUTTONS	GREEN	COOL	DIEGO
ZAPPING	STATION	EATING	PSYCHOLINGUISTICS	WALLS	COST	COMPUTER	BROTHER	REMOTE	MEAN	HANDY	BROOKLYN	REACTOR
CONSUMED	CELLS	ZEE	PROBLEM	USABLE	CONTROL	KIDS	ARM	SCROLL	FIND	BLACK	MEMBERS	BURRITOS
BAR	RECHARGEABLE	GIRAFFE	LOOSEY	PROFILE	POINT	DISTRICT	LIVING	L_C_D_	CONTROLS	BANANA	BELGIUM	FLIPPING
HAPPENED	REMOTE	HIGH	LOOKS	MISTAKE	PERCENT	TECH	SOFA	MEAN	DEVICE	FRUITS	CONCERT	FUTURE
INTERESTING	DYNAMO	ROBUST	DISFLUENCIES	BLEND	PRODUCT	APPLES	FALL	CONTROL	KIND	VEGETABLE	HOPING	COFFEE
QUESTION	SOURCE	WASHED	LEFTY	COMFY	PRICE	TORONTO	WIRELESS	CHANNELS	LIGHT	CUP	SUPPOSED	GREW
R_S_I_	DOCKING	HAPPENS	PRECIES	HARD	FORTY	ASSOCIATE	STREET	PRESS	SIGNAL	ACTUALLY	EDINBURGH	HOME
TIN	BUY	CELL	BIG	LOGO	EURO	LOW	CHOPPED	T_V_	INFRARED	CUTE	ANYWAYS	KINETICS
NUMBER	HAND	POCKET	SHEEP	CHARGED	MEAN	PRETTY	PIZZA	KIND	ACTUALLY	PUMPKIN	APRIL	LIVED
RANDOM	IDEA	WASH	JA	SHUT	SELL	BAD	BIG	FUNCTIONS	DESIGN	COMES	FIFTEEN	R_C_
APPRECIATE	LIGHT	KIDDING	MOOSE	SIDE	LOOK	CHEAP	LAPTOPS	TOP	CHIP	MEAN	LOVE	WORKS
FOIL	SURE	OUCH	ELK	INDICATOR	MONEY	FIGURE	SPECIFICALLY	SIDE	IDEA	BASE	'FRAID	COUNT
MINIMISE	MEAN	MATERIAL	TIGHTY	UNDERNEATH	TARGET	BOUGHT	HORRIBLE	HAND	TELETEXT	SEASON	NICE	BAD
ACCIDENT	KIND	CHEW	RIGHTY	HOLE	GROUP	BOTHERS	GLOB	WHEEL	LOST	RUNNING	SOUNDS	CALIFORNIA
TRIPLE	CHIP	FACE	LOVE	COLOUR	LOSE	MINUS	SWAHILI	PITCH	MANDO	PLAY-DOH	RECOGNITION	DESIGN
BANANA	SCROLL	REMOTE	MUSIC	COLOURS	HARD	IRRITATING	LEGOS	DEAR	SPANISH	SMELLS	SPEECH	MEETING
COUNT	CURVED	PLATES	SMALLVILLE	REMOTE	PEN	COBRA	WRITTEN	FLIP-OVER	HOMER	NICE	REMOTE	PROJECT
DASH	BUTTONS	BREAK	WRITE	RUBBER	FINGERS	CHRISTA	SPACESHIP	MONO	LAPEL	EDIBLE	VOICE	MINUTES
REMOTE	WHEEL	BUY	MAIN	YELLOW	SIXTY	GUYS	SPACESHIPS	TONE	SOUNDS	GUESS	L_C_D_	USER
WRITE	SPECIAL	DUNNO	WEEKS	KIND	PERSONALLY	PROTOTYPE	CHINESE	TONES	NINJA	FUNNY	CONTROL	INTERFACE
R_E_A_	BUTTON	CREATE	BIG	MEAN	FUNCTION	BANANA	SOFTWARE	LIVE	ENGLISH	TAIL	EASY	PRESENTATION
R_C_	SINGLE	MOBILE	COUPLE	FRUIT	BUILT	SMALL	FRENCH	STEREO	LOGO	WRITE	LOOK	LOOK
REAL	RUBBER	EXACTLY	FAIR	SHAPE	FIELDS	DEFINITELY	TRUE	BELIEVE	MANGO	STUFF	FANCY	PRODUCT
A_S	CURVE	PLATE	SHOW	BUTTONS	GREATER	ANNOYING	INTERNATIONAL	FREQUENCY	M_A_	GUY	MEAN	SURE
BONGIORNO	CASE	COOL	GREATER	SPONGY	ACCOMPANYING	ORANGE	ACCOMMODATED	HEIGHT	COOL	FAMILY	TECHNOLOGY	BIT
REACTION	L_C_D_	STARTED	T_V_	BLACK	DEMANDS	STRONG	ICONS	PRECISE	LOT	FORGOT	BUTTONS	DRAW
SURE	COLOUR	KIND	'EM	CONTROL	LOST	SURVEY	JA	WANTED	FUNNY	TRUE	DESIGN	THANK
PUNISH	REGULAR	CHOICE	TERRIBLE	PLASTIC	QUESTION	INTERNET	ENGLISH	TALK	HMMM	INCLUDING	SCREEN	GUESS
SHORT	ADVANCED	PRODUCTS	ASSUMING	LOOK	ADJUST	HIGHLY	DUTCH	CALM	SPAM	ACTUALLY	IMPORTANT	EVALUATION
CATCHY	PUSH	MARTIAN	FAN	BIT	DIFFICULT	TEAM	SPEAKER	FLIPPER'S	MARKET	FOLK	COST	START
OPTION	DOUBLE	T_V_	LIKES	BANANA	ZOOM	SCRATCHING	LANGUAGES	EASIER	SELLING	MULTI-COLOURED	INNOVATIVE	ACTUALLY
SCRATCH	SAMPLE	MARKETING	TELEVISION	FEEL	SET	FIGHTING	LANGUAGE	PROGRAM	U_S_	SINGER	POINT	WORKING
THINKING	PLASTIC	FASHIONABLE	RUBBISH	WOOD	NEE	FALLINGS	REALITY	DIGESTING	BECASUE	PLAY-DOHS	KIND	MARKETING
FILL	EXPENSIVE	PHONE	LESSER	RED	MICROPHONE	GOUGES	TRYING	UNCOMFORTABLE	SEND	LAME	CHIP	IDEAS

Table D.3: Example topics by the roleHDP on fold 2–4 of the AMI Meeting Corpus.

BUTTON	CHIP	MACHINERY	BRUSSELS	INTERFACE	STRAIN	LAPEL	FRUIT	TWENTY	SCROLL	COBRA	JA	FLIPPING	SCREEN
BUTTONS	ADVANCED	HIGH	HMMM	COMMAND	INJURY	HMMM	REMOTE	TWELVE	WHEEL	CHRISTA	PRECIES	KINETICS	MINUS
CHANNEL	REGULAR	MODES	COOL	EXPERT	REPETITIVE	STRAWBERRY	SPONGY	EUROS	BUTTON	SMALL	ORIGINS	EVENTUALLY	CARD
SCREEN	SPEAKER	EDITOR	AFRO	COMPONENT	R_S_I_	MIC	BANANA	POINT	PUSH	INTERNET	TRICK	THROWING	CUT
VOLUME	SAMPLE	LYING	BEAT	MARKETING	CLICKING	CROTCH	VEGETABLES	FIFTY	BUTTONS	EXACTLY	ENGLISH	COUNT	FORTY
MENU	INFRARED	IDLE	ORCHESTRA	EBENEZER	TEN	SHIRTS	VEGETABLE	COST	DISPLAY	BROADBAND	RUSSIAN	STRONG	AUDIENCE
REMOTE	SIMPLE	SIMILARLY	EDINBURGH	CONTENT	USERS	DAUGHTER	FASHION	PRICE	L_C_D_	SNAKE	ORDER	SHAKING	TARGETED
L_C_D_	CIRCUIT	CONTROLLED	APRIL	FORM	START	WORKING	KIND	EURO	INTEGRATED	MODEMS	ADJUST	BREAKS	PRODUCT
CHANNELS	BATTERY	MOUSE	BROOKLYN	FASHIONABLE	O'CLOCK	WHITE	MEAN	MARKET	SPECIAL	PAST	STRANGE	EYEBALL	IMPORTANT
SCROLL	PRINT	V_I_	BELGIUM	LOST	PURSE	BAD	FRUITS	SELL	PUSHBUTTON	PAVLOV	PRESENTATION	VOTES	T_F_T_
CONTROL	REMOTE	MICHAEL	CONCERT	POWERPOINT	FREQUENCY	BOARD	CONTROL	PRODUCT	LOSE	RECOGNISABLE	E_I_E_	SHIPPED	ED
T_V_	SENSOR	MATERIAL	LOVE	P_H_D_	LAUGH	PLUG	SHAPE	MEAN	L_C_	WORM	CHIP	DEFAULT	PROBLEM
PRESS	EXPENSIVE	SMART	MEMBERS	FUNCTIONALITY	IRRITATING	BONUS	MANGO	FORTY	ACTUALLY	FISH	USERS	TRACK	SIZE
MEAN	BUTTONS	SPECIFIC	FIFTEEN	REMEMBER	MINIMISE	LOST	TRENDS	REMOTE	PUSH-BUTTON	DUDE	ADDRESS	BURNS	BRAND
FUNCTIONS	BUTTON	QUALITY	ANYWAYS	USER	HIGHLY	GOD	TREND	HALF	SUPPLEMENTS	SCARY	PRINT	VERSION	MARKETER
KIND	SIGNAL	CONTACT	HOPING	DECIDED	ZAPPING	CLOSE	LOOK	SIXTEEN	POINT	CHRIS	DENNI	SHIPPING	FLOP
TELETEXT	T_V_	PROPERTIES	SUPPOSED	PSYCHOLOGY	SITTING	FALLS	COLOUR	PRODUCTION	INTERFACE	EIGHTIES	PROBLEM	AWESOME	FRONT
TURN	COST	HEAT	'FRAID	LINE	QUARTER	EASILY	DUNNO	MILLION	BAD	GUESS	STRETCH	PLAYER	EXACTLY
SWITCH	ENERGY	DESTRUCT	SOUNDS	CALLED	INJURIES	WORKED	YELLOW	FIFTEEN	CHIP	UBIQUITOUS	TECHNICAL	MEDIA	HEAT
POWER	BOARD	DESCRIBE	NICE	SHINE	IRRITATE	DIGESTING	COOL	THIRTY	COLOUR	MEAN	CURIOUS	DAVID	PROMOTE
REMOTE	TRIPLE	BATTERY	HAND	CURVED	COLOUR	COUNT	NICE	DRAW	DESIGN	PEPPER	PITCH	SNAIL	BONGIORNO
CONTROL	REAL	BATTERIES	SORT	SINGLE	COLOURS	BLACK	MISS	PEN	MEETING	GREEN	STEREO	SHELL	PUNISH
RECOGNITION	BANANA	SOLAR	MEAN	DOUBLE	RUBBER	RUNNING	BURRITOS	ANIMAL	PROJECT	GUY	MONO	TEXT	P_M_
SPEECH	REACTION	ENERGY	KIND	CURVE	BUTTONS	SPORTS	SEASON	CAT	MINUTES	FAMILY	SPORTS	SYMBOLS	ROO
T_V_	REMOTE	KINETIC	SHAPE	FLAT	YELLOW	FILL	SAN	BIT	USER	WRITE	TONES	RAPID	INTERESTING
MEAN	R_C_	POWER	BIT	CASE	SPECIAL	WEATHER	DIEGO	FAVOURITE	PRESENTATION	NICE	FREQUENCY	FUNNY	LATE
VOICE	CONTROL	REMOTE	BUTTONS	SPEAKER	PLASTIC	TEXT	GREW	PRETTY	INTERFACE	C'EST	HEIGHT	SLOW	POSSIBILITIES
LOOK	DASH	STATION	REMOTE	SURE	REMOTE	SCRAP	COFFEE	BOARD	PRODUCT	HOUSEHOLD	EVENT	KNOWN	WELCOME
BUTTONS	REACTOR	CELLS	SIDE	MEAN	MEAN	WHITE	CALIFORNIA	ACTUALLY	SURE	CA	TONE	EFFECT	WISH
L_C_D_	WRITE	RECHARGEABLE	HOLD	SAMPLE	CASE	GUIDE	WORKS	LOOKS	LOOK	CAPSICUM	BELIEVE	COMIC	THINKING
EASY	FUTURE	DYNAMO	IDEA	SIDE	BLACK	BLAND	HOMETOWN	DOG	EVALUATION	OUI	MAJOR	OYSTER	
FIND	SOUND	HAND	BIG	CURVES	KIND	NEWS	SPORTS	WRITE	THANK	POIVRON	TALK	HOLDER	
DESIGN	SURE	LIGHT	SMALL	DOUBLE-CURVED	WOOD	KIND	LIVED	DIGITAL	MARKETING	FOLK	POINTING	MARKETING	
CONTROLS	LOOKS	SOURCE	SIZE	RUBBER	MATERIAL	LOOKING	POUNDS	WHITEBOARD	START	SINGER	LAUNCH	FEATURE	
KIND	R_E_A_	BUY	LEFT	UNCURVED	SPONGY	DIRECT	ROBUST	DUNNO	REMOTE	FRENCH	TESTING	PRINTED	
LOT	OPTION	CONTROL	HANDED	SINGLE-CURVED	CONTROL	TICKER	COST	DRAWING	WORKING	LAME	OPTION	SURE	
MARKET	CATCHY	IDEA	DUNNO	IDEA	TITANIUM	ENTIRE	FREE	NICE	GUESS	PUMPKIN	ADDITIONAL		
FANCY	SHORT	SURE	TOP	POINT	LOOK	NUMEROUS	CUP	LOOK	DISCUSS	LOCAL	JANUARY		
TELEVISION	DELUXE	MEAN	THUMB	KIND	SHAPE	COLUMN	LISTENING	GUESS	IDEAS	MICHIGAN	FOOTBALL		
IMPORTANT	BASS	DOCKING	CONTROL	CHANGE	RED	BOTTOM	SOCCER	COOL	DESIGNER	BASICALLY	END		

Table D.4: Examples of top 28 topics by the LDA-based topical n-gram model on the AMI Meeting Corpus.

12907	12746	12679	12662	12648	12606	12601	12559	12555	12549	12503	12497	12487	12475
KIND	BUTTONS	BUTTON	MEAN	MEETING	L_C_D_	CONTROL	DESIGN	FUNCTIONS	BUTTONS	SCREEN	REMOTE	HMMM	BUTTONS
CASE	MEAN	KIND	BUTTON	COLOUR	FIND	CURVED	BUTTON	PROJECT	DESIGN	SHAPE	EASY	PROBLEM	LOT
ROOM	STUFF	TELEVISION	PRETTY	POINT	SIDE	TECHNICAL	SIMPLE	REMOTE	VOLUME	MATERIAL	POINT	BIG	EXACTLY
IMPORTANT	CONTROL	IMPORTANT	CHANNEL	SMALL	SCREEN	LOOK	FIFTY	CHIP	BIT	DUNNO	LEFT	THOUGHT	DESIGN
CONTROLS	FEATURES	WORKING	LOOKING	DESIGN	EASY	GROUP	PLASTIC	COMPANY	FASHION	CONTROL	FORTY	PUSH	BIT
LOT	THINKING	CONTROL	CONTROL	INTERNET	BUY	CURRENT	DISCUSS	EXPENSIVE	MAIN	BUTTON	CONTROL	DECISION	RECOGNITION
TELEVISION	NICE	PRESENTATION	DESIGN	TALKING	IDEAS	BASICALLY	SCREEN	BUTTONS	IDEA	HAND	USERS	DECIDE	CONTROL
COMPONENTS	PRESS	SET	INTERFACE	BUTTON	CONTROL	RUBBER	MARKETING	EUROS	CONTROL	DECIDE	BIT	INSIDE	TELETEXT
DISPLAY	GUESS	FOUND	STICK	INTERFACE	SHAPE	CIRCUIT	HOLD	BANANA	IMPORTANT	INTERFACE	NICE	REMOTE	TITANIUM
FEEL	OPTION	INTERFACE	BASICALLY	CONCEPT	SLIDE	DEVICE	ENERGY	SURE	REAL	PART	INTERFACE	HALF	MATERIALS
LATER	POINT	POINT	ADVANCED	FASHION	EUROS	DOCUMENTS	MEANS	PEN	INFRARED	TWELVE	ANIMAL	QUESTION	MEETING
TEN	EXPENSIVE	CHANNEL	BIG	MONEY	SENSE	DISPLAY	IDEA	HMMM	COOL	PRODUCT	RECOGNITION	INNOVATIVE	LEARN
IDEA	REQUIREMENTS	YELLOW	METHOD	TECHNOLOGY	DECIDED	YOUNG	MINUTES	PROTOTYPE	PLASTIC	CIRCUIT	TRUE	RESEARCH	MILLION
CONTROLLER	SIMPLE	DEFINITELY	CONCEPT	WHEEL	RUBBER	START	BASICALLY	DESIGN	BUDGET	COMES	BUTTONS	DESIGNED	USERS
COST	ABLE	RECOGNITION	DRAW	DISPLAY	CHIP	YELLOW	BUTTONS	D_V_D_	SPECIAL	ROUND	STUFF	PHONE	COMPANY
SURE	HOPE	TEAM	LOOKS	MEANS	TWENTY	ENERGY	TRENDS	COVERS	TEN	POINT	THINKING	ELDERLY	COOL
SHAPE	SCROLL	DISCUSS	COLOUR	LOT	SIZE	EXPENSIVE	REMOTES	COMING	TELL	CHIP	DEPENDS	FEEL	OPTION
SOURCE	SLIDE	CHANNELS	SIDE	REMOTE	CRITERIA	HMMM	TOUCH	PERCENT	TECHNICAL	CUSTOMERS	MEETING	SUPPOSE	COLOURS
TELETEXT	MATERIALS	COURSE	FUNCTIONAL	PHONES	FUNCTIONS	NEEDS	PERSONAL	PART	HMMM	COST	CALL	MEETING	NUMBER
SHOW	DESIGN	MAIN	EXACTLY	OPINION	MESSAGES	COMPANY	STANDARD	SIGNAL	CHIP	L_C_D_	FEEL	DESIGN	PERSONALLY

12474	12470	12443	12429	12408	12390	12389	12371	12364	12284	12269	12263	12221	12220
REMOTE	TOP	SPEECH	L_C_D_	REMOTE	CONTROL	LOOK	RUBBER	LIGHT	USER	MARKET	TELEVISION	REMOTE	KIND
MEAN	CONTROL	REMOTE	DEVICE	ACTUALLY	PRODUCT	SCROLL	POWER	CHANGE	SORT	BUTTONS	BUTTON	MEAN	ADD
FANCY	IDEA	COLOURS	FUNCTIONS	TALK	TWENTY	END	CHOOSE	DUNNO	CONTROL	GUESS	BIT	USER	MEETING
CONTROL	DESIGN	KIND	BUTTONS	BATTERY	CHANNELS	BUTTON	CONTROLS	EASY	BASICALLY	BASIC	LOOKS	FUNCTION	PROJECT
IDEA	SURE	T_V_	BLACK	TERMS	MARKET	GUESS	SPONGY	PERCENT	T_V_	CONTROL	WATCH	CONTROL	PRICE
GUYS	CHIP	SORT	LOOKING	TURN	CHANGE	PROJECT	MOVE	CONTROLS	START	SWITCH	FEEL	MEETING	CONTROLS
LOOKS	HALF	THOUGHT	COURSE	BIT	ENERGY	SURE	DESIGN	BOARD	RECOGNITION	COMPONENTS	QUESTION	VOICE	PART
BIG	PERCENT	PRESS	BATTERIES	MARKETING	VOICE	HAND	LOT	IDEA	IMPORTANT	MENU	GUYS	MIDDLE	SET
VOLUME	MENU	MEETING	INTERFACE	NORMAL	WHEEL	OPTIONS	REQUIREMENTS	COST	LOOKS	DOUBLE	FOLDER	BIT	FAIRLY
ABLE	CONCEPTUAL	WORKING	DISCUSSED	STATION	BASIC	VOLUME	SPEECH	ABLE	THIRTY	ASPECT	PROBLEM	PRESS	CHANNEL
COSTS	OPTIONS	END	FANCY	INTERFACE	FIFTY	NUMBERS	KIND	MINUTES	MENU	EASIER	OPEN	REMEMBER	BEEP
BIT	LEAVE	COMPLICATED	PRICE	TECHNOLOGY	THOUGHT	SCREEN	SORT	FRUIT	KINDA	TALKING	MEAN	COST	FOCUS
INTERESTING	HMMM	SETTINGS	COVER	MOMENT	PLASTIC	TRENDY	YOUNGER	CASE	BOTTOM	MIND	TIMES	YELLOW	AGREE
CONTROLS	UNIVERSAL	PRETTY	TELEVISION	COLOUR	BATTERY	IDEAS	FRUIT	MOBILE	IDEA	NEEDS	DIFFERENCE	DESIGNER	EURO
SLOGAN	TYPE	WHEELS	WRITE	PUSH	NUMBER	INDUSTRIAL	FUNCTION	PRODUCTION	MIND	REMOTE	REQUIREMENTS	THANK	FIT
DESIGN	DRAW	TALKING	MANAGEMENT	DESIGNER	MONEY	EXPERT	WHITEBOARD	PROFIT	SPEAKER	EURO	HARD	POWER	NUMBER
EUROS	COMPONENTS	ANIMAL	PLACE	ACTUAL	SEND	THINKING	WILLING	AGENDA	FIND	FACT	BASE	PRESENTATION	THINKING
FRONT	SAYING	SOLAR	JOYSTICK	SOURCE	RECOGNITION	COLOURS	WHEEL	EASILY	PROGRAMME	SCREEN	MATCH	HOUR	PROTOTYPE
LOT	EVALUATE	TWENTY	MENU	HMMM	CURVED	FRIENDLY	SURE	EMAIL	FUNCTIONAL	HAND	CONFUSING	BASICALLY	THUMB
ORIGINAL	TECHNOLOGY	USERS	KIND	CHOOSE	COST	WATCHING	CONTROL	THIRTY	EURO	FEEL	FUNCTIONAL	SAYING	SPECIAL

Table D.5: Examples of top 28 topics by the HDP-based topical n-gram model (HDPLM) on the AMI Meeting Corpus.

38160	33808	27508	26925	22963	19289	15612	13310	11911	10899	10789	10407	8908	8892
FIND	STUFF	BUTTONS	DESIGNER	BUTTONS	SORT	ABLE	REMOTE	DISCUSS	COMPANY	RUBBER	FEEL	SPEECH	LOOKS
LOOK	USER	KIND	IMPORTANT	SCREEN	IDEA	SURE	DESIGN	START	RESEARCH	SPONGY	VEG	VOICE	LOOK
PRESS	MARKET	FUNCTIONS	CONCEPT	INTERFACE	WATCHING	TALKING	MEETING	BUY	EXPERT	TITANIUM	BRING	L_C_D_	BUTTON
CHANGE	YELLOW	COLOUR	EUROS	SCROLL	PREFERENCES	EASY	T_V_	TALK	TECH	HANDED	CASE	MAIN	ANIMAL
CHOOSE	EASY	COLOURS	PLAYER	MENU	FANCY	THINKING	PROJECT	ADD	AHEAD	WORKING	OBVIOUSLY	USER	POINT
SELL	VEGETABLE	ENERGY	PLAYERS	SIDE	SIMPLE	TRYING	USER	LOOK	FOCUS	HAND	CHANGE	TELETEXT	HMMM
HOLD	SIMPLE	POWER	T_V_S	DISPLAY	TELEVISION	SUPPOSED	PRESENTATION	SPEND	MACHINE	INDIVIDUAL	MEANS	SPECIAL	EXACTLY
TURN	SIXTY	USERS	VIEW	PROBLEM	GROUP	TRUE	PRODUCT	CONSIDER	SHARE	LIGHTS	KIND	CHANNEL	SHAPE
TELL	EXACTLY	FEATURES	PAGE	HAND	ROUND	LOOKING	TELEVISION	INNOVATIVE	UNDERSTAND	PLASTIC	MEAN	BIG	SORT
MOVE	OBVIOUSLY	MATERIAL	REMOTE	NUMBER	EXTRA	PRETTY	IDEA	ASK	TOUCH	CHANNEL	MAKES	FRUIT	FEATURE
DRAW	ACCOUNT	OPTIONS	HONEST	DEVICE	ORIGINAL	LOST	CHANNEL	STICK	SURVEY	TURN	FINALLY	INDUSTRIAL	PROJECT
SEND	YOUNGER	TECHNOLOGY	TYPE	FUNCTION	COACH	WILLING	END	SWITCH	PRICE	BASICALLY	IMPORTANT	YOUNG	BASED
PAY	TELETEXT	RUBBER	LIKELY	REMOTE	REAL	GETTING	IDEAS	LEARN	QUICKLY	SOUND	SPEECH	INTERNET	CHANNELS
REMEMBER	BLACK	BATTERY	INTEGRATED	FRIENDLY	LOTS	SAYING	FACT	LEAVE	FIGURE	NORMAL	YOUNGER	HALF	YELLOW
LOSE	SAMPLE	SHAPE	HUGE	ASPECT	FLAT	FINE	PERSONAL	THIRTY	NICHE	BASIC	WHISTLES	KINETIC	FORWARD
AGREE	INCORPORATE	INFORMATION	AUDIENCE	COMPONENTS	NOTES	EASIER	SET	LOSE	PLACE	HMMM	BUTTON	CASE	VOLUME
SLIDE	CURRENT	REMOTES	FOCUSED	BATTERY	SAYING	DYNAMO	FAVOURITE	SHOW	UPPER	OPTION	APPARENTLY	YELLOW	BEEP
FIT	ESPECIALLY	PRICE	USERS	DECISION	OPINION	READY	COMPANY	IMAGINE	IMPLEMENT	BRIGHT	PAINT	TECHNICAL	MAKES
BREAK	INFRARED	BATTERIES	MARKETING	QUESTIONS	TOTALLY	LEFT	MIND	MEET	ACCESS	CHEAPER	DISCUSS	REAL	STUFF
INCORPORATE	FIND	CASE	RANGE	REQUIREMENTS	MORNING	DESIGNING	FASHION	CALL	UNIVERSAL	STARTS	CRITERIA	D_V_D_	MEETING

8601	7711	7558	7421	6901	6494	6478	6254	5912	5631	5208	3463	2704	2085
MONEY	BASICALLY	DECIDE	HMMM	FOLDER	WHEEL	MEETING	ACTUALLY	KIND	COURSE	MARKET	BUTTON	BIT	CONTROL
IMAGE	REMOTE	VOLUME	IMPORTANT	MANAGER	GUYS	MEANS	PUSH	PRETTY	T_V_	USER	MINUTES	LOT	MEAN
RESEARCH	IDEA	TWENTY	EASY	PLAN	WANTED	NEEDS	SORT	LOOKING	BUTTONS	COST	REMOTE	L_C_D_	RECOGNITION
WORKING	FIFTY	TOP	EXPENSIVE	INSIDE	WHEELS	COMES	DEFINITELY	KINDA	THINKING	PRODUCT	POINT	CHIP	CONTROLS
BEHAVIOUR	ACTUALLY	REMOTE	NICE	REQUIREMENTS	FEEL	SORT	SEEN	ACTUALLY	SURE	BOARD	PERCENT	DEVICE	GUESS
WIDTH	CHANNEL	LOOK	BIG	THANK	REQUIRE	LATER	KIND	FANCY	TWELVE	BASIC	BUTTONS	BUTTON	DUNNO
BULB	FEEL	FEEL	SIMPLE	EVALUATION	CLOSE	MAKES	LOST	BASICALLY	THIRTY	EVALUATION	EUROS	LOOK	THOUGHT
PRODUCTION	THANK	CHANNELS	SMALL	PERSON	RECEIVE	WANTS	DECIDED	SWITCH	RUBBER	ACTUAL	EURO	IDEA	SUPPOSE
MATERIAL	WELCOME	PRINT	CURVED	WORKS	GAVE	WORKS	FOUND	WORKING	ELECTRONICS	PROTOTYPE	HUNDRED	TELEVISION	PHONE
AGREE	HELD	TELEVISION	COOL	MEANS	CLICK	KIND	BASICALLY	NICE	MIDDLE	COSTS	FIFTY	BANANA	PHONES
SET	NUMBER	TEN	HARD	SCHEME	CHANGE	DEPENDS	SHOW	THOUGHT	FORTY	CASE	MILLION	REGULAR	HOPE
WEIRD	MEETING	SENSE	INTERESTING	YELLOW	WAIT	ISSUE	DISCUSSED	DECIDED	PLASTIC	GROUP	CHANNEL	ADVANCED	FOUND
SORTS	PART	FIFTY	COMPLICATED	THANKS	PREFER	SIMPLE	TALKED	LOOK	SCROLL	YOUNGER	SIMPLE	TOP	BELIEVE
WEBSITE	FUNCTION	L_C_D_	FANCY	DOCUMENTS	DISCUSSED	OPEN	OPEN	INSIDE	EXACTLY	CHANNELS	PRESENTATIONS	DARK	INJURY
SCORE	BOARD	'EM	DIFFICULT	CERTAINLY	WATCH	SIDE	GOTTA	NECESSARILY	TWENTY	INFRARED	CHANNELS	T_V_	PUSH
FAMILIAR	TELETEXT	CHANNEL	TRENDY	HARD	DIVISION	ESPECIALLY	TOLD	WRITE	CHANNEL	PROCESS	TWENTY	MOBILE	WROTE
STRIPES	KIND	VEGETABLES	CHEAP	WAIT	WANTS	CASE	SAVE	STATION	TEN	DOCUMENTS	TIMES	OPTION	FORGOT
ENVIRONMENT	ESPECIALLY	POWER	MONEY	FLAT	LIKED	SOUNDS	LOOKED	SENSE	INFRARED	LOGO	SELECTION	MARKETING	RECOGNISER
METAL	DEPENDING	INFRARED	HIGH	MAINLY	LOOK	TRAINING	WORKS	WORTH	WHITEBOARD	WEB	VOLUME	BASIC	COMMAND
I_D_	SINGLE	R_S_I_	CURVE	EVALUATED	NEEDED	SENDS	THOUGHT	ASKED	INSTANCE	STATION	NUMBERS	COUPLE	ACTIVATION

Appendix E

Program and Scripts for the HPYLM and the HDP

The executable programs (hpylm and hdp), the corresponding Python scripts, and part of the text data used in this thesis to train and test an HYPLM and an HDP, are available from `http://homepages.inf.ed.ac.uk/s0562315/`. Suppose the training data is amicorpus.train.data.gz, the testing data is amicorpus.test.data.gz, and the vocabulary is amicorpus.vocab. The following are some example commands to train HPYLM and HDP models.

E.1 hpylm

Train an HPYLM using hpylm

```
[cwd@localhost]: hpylm -debug 0 -order 3 -vocab amicorpus.vocab
-pydiscount -numiter 100 -numsamp 10 -numgap 5
-text amicorpus.train.data.gz -lm amicorpus.hpylm.arpa.gz
-write amicorpus.hpymodel -ppl amicorpus.test.data.gz
```

Train an PLDLM using hpylm

```
[cwd@localhost]: hpylm -order 3 -vocab amicorpus.vocab
-text amicorpus.train.data.gz -read-with-mincounts -ebdiscount
-gt1min 1 -gt2min 1 -gt3min 1 -lm amicorpus.pldlm.arpa.gz -debug 1
-ppl amicorpus.test.data.gz -eb-use-theta
```

E.2 hdp

Train a roleHDP using hdp

```
[cwd@localhost]: hdp -train train.doc -test test.doc -config config
-numtopic 50 -numiter 3000 -numsamp 25 -numspace 100 -aa 0.5
-write-hdp foldtrain-50-0.5.rhdp -debug 1

### format of config file
[cwd@localhost]: cat config
ROOT BASE
PM ROOT
ID ROOT
UI ROOT
ME ROOT

### format of training and testing data
[cwd@localhost]: head train.doc
<role> PM </role> <doc> KICKOFF MEETING PROJECT TWENTY
MINUTES KIND SURE LAURA PROJECT MANAGER INTRODUCE </doc>
<role> UI </role> <doc> DESIGNING REMOTE CONTROL RECORD
ACTUALLY DAVID ANDREW CRAIG ARRIVED DESIGN REMOTE
CONTROL SUPPOSED ORIGINAL TRENDY USER FRIENDLY KIND
STAGES DESIGN SURE GUYS RECEIVED EMAILS </doc>
```

Infer role information using hdp

```
[cwd@localhost]: hdp -read-hdp foldtrain-50-0.5.rhdp
-test test.doc -infer-role -numspace 100 -numsamp 2
```

Bibliography

Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing1. Combining stochastic block models and mixed membership for statistical network analysis. In *Statistical Network Analysis: Models, Issues, and New Directions*, Lecture Notes in Computer Science, pages 57–74. Springer-Verlag, 2007.

Yuya Akita, Yusuke Nemoto, and Tatsuya Kawahara. PLSA-based topic detection in meetings for adaptation of lexicon and language model. In *Proc. of Interspeech*, August 2007.

David Aldous. Exchangeability and related topics. In *École d'été de probabilités de Saint-Flour*, XIII–1983, pages 1–198. Springer, Berlin, 1985.

Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1):5–43, January 2003.

Charles. E. Antoniak. Mixtures of Dirichlet processes with application to non-parametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

Matthew P. Aylett. Detecting high level dialog structure without lexical information. In *Proceedings of ICASSP'06*, Toulouse, France, May 2006.

L. R. Bahl, F. Jelinek, and L. R. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:179–190, 1983.

R. F. Bales. *Interaction Process Analysis: A Method for the Study of Small Groups*. Addison-Wesley, Cambridge MA, 1951.

Kobus Barnard, Pinar Duygulu, David A. Forsyth, Nando de Freitas, David M. Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

L. E. Baum. An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities*, 3:1–8, 1972.

M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The infinite hidden Markov model. In *Advances in Neural Information Processing Systems*, volume 14, 2002.

Jerome R. Bellegarda. Exploiting latent semantic information in statistical language modeling. 88(8):1279–1296, 1998.

Jerome R. Bellegarda and John W. Butzberger. A novel word clustering algorithm based on latent semantic analysis. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 172–175, 1996.

Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13*, pages 933–938, 2001.

Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

J. A. Bilmes and K. Kirchhoff. Factored language models and generalized parallel backoff. In *Proceedings of HLT/NACCL*, pages 4–6, 2003.

David M. Blei and John D. Lafferty. Correlated topic models. In *NIPS*, 2005.

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

John Blitzer, Kilian Weinberger, Lawrence Saul, and Fernando Pereira. Hierarchical distributed representations for statistical language modeling. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 18*, Cambridge, MA, 2004. MIT Press.

John Blitzer, Amir Globerson, and Fernando Pereira. Distributed latent variable models of lexical co-occurrences. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1991.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

Wray Buntine. Variational extensions to EM and multinomial PCA. In *13th European Conference on Machine Learning (ECML'02)*, pages 23–34, Helsinki, Finland, 2002.

Wray Buntine and Aleks Jakulin. Discrete components analysis. In C. Saunders, M. Grobelnik, S. Gunn, and J. Shawe-Taylor, editors, *Subspace, Latent Structure and Feature Selection Techniques*, pages 1–34. Springer-Verlag, 2006.

Ellen Campana, Jason Baldridge, John Dowding, Beth Ann Hockey, Roger W. Remington, and Leland S. Stone. Using eye movements to determine referents in a spoken dialogue system. In *Proceedings of Workshop on Perceptive User Interface*, Orland, Florida, 2001.

John Canny. GaP: A factor model for discrete data. In *ACM Conference on Information Retrieval (SIGIR)*, pages 122–129, Sheffield, England, July 2004.

J. Carletta. Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation Journal*, 41(2):181–190, 2007.

Oscar Chan and Roberto Togneri. Prosodic features for a maximum entropy language model. In *Proceedings of Interspeech (ICSLP'2006)*, Pittsburgh, US, September 2006.

Ciprian Chelba and Frederick Jelinek. Exploiting syntactic structure for language modeling. In *Proc. of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 225–231, 1998.

Ciprian Chelba and Frederick Jelinek. Extensions to constraint dependency parsing for spoken language processing. *Computer Speech and Language*, 14(4):283–332, 2000.

Ken Chen and Mark Hasegawa-Johnson. Improving the robustness of prosody dependent language modelling based on prosody syntax dependence. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 435–440, St. Thomas, U.S. Virgin Islands, 2003.

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999.

Neil Cooke and Martin Russell. Using the focus of visual attention to improve spontaneous speech recognition. pages 1213–1216, 2005.

Philip J. Cowans. Information retrieval using hierarchical dirichlet processes. In *In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (p*, pages 564–565. ACM, 2004.

Philip J. Cowans. *Probabilistic Document Modelling*. PhD thesis, Inference Group, Cavendish Laboratory, University of Cambridge, January 2006.

S.B. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

Alfred Dielmann and Steve Renals. Recognition of dialogue acts in multiparty meetings using a switching DBN. *IEEE Transactions on Audio, Speech and Language Processing*, 16(7):1303–1314, 2008.

A. Emami and L. Mangu. Empirical study of neural network language models for Arabic speech recognition. In *Proc. of ASRU*, pages 147–152, 2007.

Ahmad Emami and Frederick Jelinek. Exact training of a neural syntactic language model. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 245–248, 2004.

Ahmad Emami and Frederick Jelinek. Random clusterings for language modeling. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 581–584, 2005.

Elena A. Erosheva, Stephen E. Fienberg, and Cyrille Joutard. Describing disability through individual-level mixture models for multivariate binary data. *Annals of Applied Statistics*, 1(2):502–537, 2007.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

E. Fox, E. Sudderth, M. I. Jordan, and A. Willsky. An HDP-HMM for systems with state persistence. In *Proc. of the 25th International Conference on Machine Learning (ICML)*, pages 312–319, 2008.

Brendan J. Frey and Nebojsa Jojic. A comparison of algorithms for inference and learning in probabilistic graphical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1392–1416, September 2005.

S. Furui. Speaker independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 34(1): 52–59, 1986.

Jianfeng Gao, Joshua Goodman, Guihong Cao, and Hang Li. Exploring asymmetric clustering for statistical language modeling. In *Proc. of 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 183–190, 2002.

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, London, second edition, 2004.

Mark Girolami and Ata Kabán. On an equivalence between PLSI and LDA. In *Proceedings of ACM SIGIR*, 2003.

Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. In *Advances in Neural Information Processing Systems 17*, pages 497–504, 2004.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 673–680, 2006a.

Sharon J. Goldwater, Thomas L. Griffiths, and Mark Johnson. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*, 2006b.

I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3/4):237–264, 1953.

Joshua Goodman. Exponential priors for maximum entropy models. In *Proc. of the HLT-NAACL*, pages 305–311, 2004.

Joshua T. Goodman. A bit of progress in language modeling. *Computer Speech and Language*, pages 403–434, 2001.

Thomas L. Griffiths and Mark Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences, 101 (suppl. 1)*, pages 5228–5235, 2004.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. Integrating topics and syntax. In *NIPS*, 2004.

T. Hain, J. Dines, G. Gaurau, M. Karafiat, D. Moore, V. Wan, R.J.F. Ordelman, and S. Renals. The development of the AMI system for the transcription of speech in meetings. In *Proc. of 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, Edinburgh, UK, 2005.

Thomas Hain, Lukas Burget, John Dines, Giulia Garau, Martin Karafiat, Mike Lincoln, Jithendra Vepa, and Vincent Wan. The AMI system for the transcription of speech in meetings. In *Proc. of ICASSP'07*, pages 357–360, April 2007.

M. P. Harper and R. A. Helzerman. Extensions to constraint dependency parsing for spoken language processing. *Computer Speech and Language*, 9:187–234, 1995.

P. Heeman and J. Allen. Incorporating POS tagging into language modeling. In *Proc. of the European Conference on Speech Communication and Technology*, pages 2767–2770, 1997.

Peter Heeman. POS tagging versus classes in language modeling, 1998.

H. Hermansky. Perceptual linear predictive (PLP) analysis of speech. *Journal of the Acoustic Society of America*, 87(4):1738–1752, 1990.

Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, UAI'99*, 1999.

Thomas Hofmann and Jan Puzicha. Statistical models for co-occurrence data. In *Proceedings of the International Conference Pattern Recognition*, pages 192–194, 1998.

Andrew Wilson Howitt. *Automatic Syllable Detection of Vowel Landmarks*. PhD thesis, Massachusetts Institute of Technology, 2000.

Bo-June Hsu and James Glass. Style and topic language model adaptation using HMM-LDA. In *Proc. of EMNLP*, July 2006.

P. Hsueh and J. Moore. Automatic topic segmentation and lablelling in multiparty dialogue. In *Proc. of the first IEEE/ACM workshop on Spoken Language Technology (SLT)*, 2006.

P. Hsueh and J. Moore. What decisions have you made: Automatic decision detection in conversational speech. In *Proc. of NACCL/HLT*, pages 25–32, 2007.

Songfang Huang and Steve Renals. Hierarchical Pitman-Yor language models for ASR in meetings. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'07)*, pages 124–129, Kyoto, Japan, 2007a.

Songfang Huang and Steve Renals. Modeling prosodic features in language models for meetings. In A. Popescu-Belis, S. Renals, and H. Bourlard, editors, *Machine Learning for Multimodal Interaction IV*, volume 4892 of *Lecture Notes in Computer Science*, pages 191–202. Springer-Verlag, 2007b. doi: http://dx.doi.org/10.1007/978-3-540-78155-4_17.

Songfang Huang and Steve Renals. Unsupervised language model adaptation based on topic and role information in multiparty meetings. In *Proc. Interspeech'08*, pages 833–836, Brisbane, Australia, 2008a.

Songfang Huang and Steve Renals. Modeling topic and role information in meetings using the hierarchical Dirichlet process. In A. Popescu-Belis and R. Stiefelhagen, editors, *Machine Learning for Multimodal Interaction V*, volume 5237 of *Lecture Notes in Computer Science*, pages 214–225. Springer-Verlag, 2008b. doi: http://dx.doi.org/10.1007/978-3-540-85853-9_20.

Songfang Huang and Steve Renals. Using participant role in multiparty meetings as prior knowledge for nonparametric topic modeling. In *Proc. ICML/UAI/COLT Workshop on Prior Knowledge for Text and Language Processing*, pages 21–24, Helsinki, Finland, 2008c.

Songfang Huang and Steve Renals. A parallel training algorithm for hierarchical Pitman-Yor process language models. In *Proc. Interspeech'09*, pages 2695–2698, Brighton, UK, 2009.

Songfang Huang and Steve Renals. Power law discounting for n-gram language models. In *Proc. IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP'10)*, pages 5178–5181, Dallas, Texas, USA, 2010a.

Songfang Huang and Steve Renals. Hierarchical bayesian language models for conversational speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 18(8):1941–1954, 2010b.

Hemant Ishwaran and Lancelot F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. The ICSI meeting corpus. In *Proceedings of IEEE ICASSP*, Hong Kong, China, April 2003.

H. Jeffreys. *Theory of Probability*. Clarendon Press, Oxford, 2nd edition, 1948.

F. Jelinek. Self-organized language modeling for speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Readings in Speech Recognition*. Morgan Kaufman Publishers, Inc., 1990.

F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In E. S. Gelsema and L. N. Kanal, editors, *Proc. Workshop on Pattern Recognition in Practice*, pages 381–397. North-Holland Publishing Company, 1980.

Frederick Jelinek. Up from trigrams! - the struggle for improved language models. In *Proc. of EUROSPEECH*, pages 1037–1040, 1991.

Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1997.

Gang Ji and Jeff Bilmes. Multi-speaker language modelling. In *Proceedings of HLT-NAACL SIGDIAL Workshop*, Boston, MA, 2004.

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

W. E. Johnson. Probability: deductive and inductive problems. *Mind*, 41:421–423, 1932.

Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37 (2):183–233, 1999.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall, second edition, 2009.

S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401, 1987.

K. Kirchhoff, J. Bilmes, S. Das, N. Duta, M. Egan, G. Ji, F. He, J. Henderson, D. Liu, M. Noamany, P. Schone, R. Schwartz, and D. Vergyri. Novel approaches to arabic speech recognition: Report from the 2002 johns-hopkins workshop. In *Proc. of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 344–347, 2003.

R. Kneser, J. Peters, and D. Klakow. Language model adaptation using dynamic marginals. In *Proc. of Eurospeech*, Rhodes, 1997.

Reinhard Kneser and Hermann Ney. Improved clustering techniques for class based statistical language modeling. In *Proc. of European Conference on Speech Communication and Technology*, pages 973–976, 1993.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proc. of ICASSP'95*, pages 181–184, 1995.

David B. Koons, Carlton J. Sparrell, and Kristinn R. Thorisson. Integrating simultaneous input from speech, gaze, and hand gestures. In M. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 257–276. AAAI Press, Menlo Park, CA, USA, 1993.

R. Krauss, C. Garlock, P. Bricker, and L. McMahon. The role of audible and visible back-channel responses in interpersonal communication. *Journal of Personality and Social Psychology*, 35(7):523–529, 1977.

P. S. Laplace. *Philosophical Essay on Probabilities*. Springer, Berlin, 1995, 5th edition, 1825.

Xin Lei, Manhung Siu, Mei-Yuh Hwang, Mari Ostendorf, and Tan Lee. Improved tone modeling for mandarin broadcast news speech recognition. In *Proceedings of Interspeech (ICSLP'2006)*, Pittsburgh, US, September 2006.

Fei-Fei Li and Pietro Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR (2)*, pages 524–531. IEEE Computer Society, 2005.

Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein. The infinite pcfg using hierarchical dirichlet processes. In *In EMNLP'07*, 2007.

G. J. Lidstone. Note on the general case of the Bayes–Laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8:182–192, 1920.

David J. C. MacKay and Linda C. Bauman Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):1–19, 1994.

David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

Sven Martin, Jörg Liermann, and Hermann Ney. Algorithms for bigram and trigram word clustering. *Speech Communication*, 24(1):19–37, 1998.

Andrew McCallum, Andres Corrada-Emmanuel, and Xuerui Wang. Topic and role discovery in social networks. In *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI)*, 2005.

Peter McCullagh. What is a statistical model? *Annals of Statistics*, 30(5):1225–1310, 2002.

Paul Mermelstein. Automatic segmentation of speech into syllabic units. *Journal Acoustical Society of America*, 58(4):880–883, October 1975.

Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

Nelson Morgan. The meeting project at ICSI. In *Proceedings of the Human Language Technology Conference*, San Diego, CA, 2001.

Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *the Tenth International Workshop on Artificial Intelligence and Statistics*, 2005.

David Mrva and Philip C. Woodland. Unsupervised language model adaptation for mandarin broadcast conversation transcription. In *Proc. of Interspeech*, September 2006.

G. Murray, S. Renals, and J. Carletta. Extractive summarization of meeting recordings. In *Proc. of Interspeech*, pages 593–596, 2005.

A. Nadas. Estimation of probabilities in the language model of the IBM speech recognition system. *IEEE Trans ASSP*, 32(4):859–861, 1984.

Hermann Ney, Ute Essen, and Reinhard Kneser. On structuring probabilistic dependences in stochastic language modeling. *Computer Speech & Language*, 8(1):1–38, 1994.

Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

Sharon Oviatt. Breaking the robustness barrier: Recent progress on the design of robust multimodal systems. *Addvances in Computers*, 56:305–341, 2002.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. of 40th Annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Jim Pitman. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102:145–158, 1995.

Jim Pitman. Combinatorial stochastic processes. Technical Report Technical Report 621, Department of Statistics, University of California at Berkeley, 2002. Lecture Notes for ST. Flour Summer School.

Jim Pitman and Marc Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997.

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, pages 275–281, 1998.

Jonathan K. Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

Steve Renals, Thomas Hain, and Hervé Bourlard. Recognition and interpretation of meetings: The AMI and AMIDA projects. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU'07)*, 2007.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, Banff, Canada, July 2004.

Deb Roy and Niloy Mukherjee. Towards situated speech understanding: Visual context priming of language models. *Computer Speech and Language*, 19(2):227–248, 2005.

Martin Russell, Stephen Cox, Antje Meyer, and Alan Wing. Multiple-level models for multi-modal interaction. In *Proceedings of the Workshop on Speech, Language and Human Computer Interaction*, Cambridge, UK, June 2004.

G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

George Saon, Hagen Soltau, Upendra Chaudhari, Stephen Chu, Brian Kingsbury, Hong-Kwang Kuo, Lidia Mangu, and Daniel Povey. The ibm 2008 gale arabic speech trranscription system. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4378–4381, 2010.

Lawrence Saul and Fernando Pereira. Aggregate and mixed-order Markov models for statistical language processing. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP-97)*, pages 81–89, 1997.

H. Schwenk and J.-L. Gauvain. Using neural network language models for LVCSR. In *Proc. of 2004 Rich Transcriptions Workshop*, Pallisades, NY, 2004.

Holger Schwenk and Jean-Luc Gauvain. Connectionist language modeling for large vocabulary continuous speech recognition. In *Proceedings of IEEE International Conference on Acoustic Speech and Signal Processing (ICASSP)*, pages 765–768, May 2002.

Holger Schwenk and Jean-Luc Gauvain. Training neural network language models on very large corpora. In *Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 201–208, 2005.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

Elizabeth Shriberg and Andreas Stolcke. Direct modeling of prosody: An overview of application in automatic speech processing. In *Proceedings of International Conference on Speech Prosody*, Nara, Japan, March 2004.

Elizabeth Shriberg, Rebecca Bates, and Andreas Stolcke. A prosody-only decision-tree model for disfluency detection. In *Proc. of EUROSPEECH*, pages 2383–2386, 1997.

Elizabeth Shriberg, Andreas Stolcke, Dilek Z. Hakkani-Tür, and Gökhan Tür. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, 32(1-2):127–154, 2000. Special Issue on Accessing Information in Spoken Audio.

Elizabeth Shriberg, Andreas Stolcke, and Don Baron. Can prosody aid the automatic processing of multi-party meetings? evidence from predicting punctuation, disfluencies, and overlapping speech. In *Proceedings of ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, Red Bank, NJ, 2001.

Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision (ICCV 2005)*, 2005.

Kemal Sönmez, Elizabeth Shriberg, Larry Heck, and Mitchel Weintraub. Modeling dynamic prosodic variation for speaker verification. In *Proceedings of 5th International Conference on Spoken Language Processing*, pages 3189–3192, Sydney, Australia, 1998.

Andreas Stolcke. SRILM - an extensible language modeling toolkit. In *Proceedings of ICSLP'02*, pages 901–904, Denver, Colorado, September 2002.

Andreas Stolcke, Elizabeth Shriberg, Dilek Hakkani-Tür, and Gökhan Tür. Modeling the prosody of hidden events for improved word recognition. In *Proceedings of 6th European Conference on Speech Communication and Technology*, pages 307–310, Budapest, Hungary, 1999.

E. B. Sudderth. *Graphical Models for Visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology, Cambridge, May 2006.

Yik-Cheung Tam and Tanja Schultz. Unsupervised lm adaptation using latent semantic marginals. In *Proc. of Interspeech*, Sep 2006.

Paul Taylor, Simon King, Stephen Isard, and Helen Wright. Intonation and dialog context as constraints for speech recognition. *Language and Speech*, 41(3-4):489–508, 1998.

Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Y. W. Teh. Dirichlet processes. Submitted to Encyclopedia of Machine Learning, 2007.

Y. W. Teh, K. Kurihara, and M. Welling. Collapsed variational inference for HDP. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

Yee Whye Teh. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore, 2006a.

Yee Whye Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proc. of the Annual Meeting of the ACL*, volume 44, 2006b.

Yee Whye Teh and Michael I. Jordan. Hierarchical Bayesian nonparametric models with applications. In N. Hjort, C. Holmes, P. Müller, and S. Walker, editors, *To appear in Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.

G. Tur, A. Stolcke, L. Voss, S. Peters, D. Hakkani-Tür, J. Dowding, B. Favre, R. Fernandez, M. Frampton, M. Frandsen, C. Frederickson, M. Graciarena, D. Kintzing, K. Leveque, S. Mason, J. Niekrasz, M. Purver, K. Riedhammer, E. Shriberg, J. Tien, D. Vergyri, and F. Yang. The CALO meeting assistant system. *IEEE Transactions on Audio, Speech and Language Processing*, 18(6):1601–1611, 2010.

N. M. Veilleux and M. Ostendorf. Prosody/parse scoring and its applications in ATIS. In *Proceedings of ARPA HLT Workshop*, pages 335–340, Plainsboro, NJ, 1993.

Alex Waibel, Michael Bett, Florian Metze, Klaus Ries, Thomas Schaaf, Tanja Schultz, Hagen Soltau, Hua Yu, and Klaus Zechner. Advances in automatic meeting record creation and access. In *Proc. of IEEE ICASSP*, Salt Lake City, UT, May 2001.

Alexander Waibel and Rainer Stiefelhagen, editors. *Computers in the Human Interaction Loop*. Human-Computer Interaction Series. Springer, 2009.

Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

Vincent Wan and Thomas Hain. Strategies for language model web-data collection. In *Proc. of ICASSP'06*, pages 1069–1072, Toulouse, France, May 2006.

Wen Wang and Mary P. Harper. The SuperARV language model: Investigating the effectiveness of tightly integrating multiple knowledge sources. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 238–247, 2002.

Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proc. of the 7th IEEE International Conference on Data Mining*, pages 697–702, 2007.

Kilian Q. Weinberger, Fei Sha, and Lawrence K. Saul. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the Twenty First International Conference on Machine Learning (ICML-04)*, Banff, Canada, 2004.

Ian H. Witten and Timothy C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, 1991.

F. Wood and Y. W. Teh. A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 12, pages 607–614, 2009.

Jun Wu and Sanjeev Khudanpur. Building a topic-dependent maximum entropy language model for very large corpora. In *Proceedings of ICASSP2002*, Orlando, USA, May 12-17 2002.

E. P. Xing, K. Sohn, M. I. Jordan, and Y. W. Teh. Bayesian multi-population haplotype inference via a hierarchical Dirichlet process mixture. In *Proc. of the 23rd International Conference on Machine Learning (ICML)*, 2006.

E. P. Xing, M. Jordan, and R. Roded. Bayesian haplotype inference via the dirichlet process. *Journal of Computational Biology*, 14(3):267–284, 2007.

Peng Xu and Frederick Jelinek. Using random forests in the structured language model. In *Advances in Neural Information Processing Systems*, 2004.

Peng Xu, Ahmad Emami, and Frederick Jelinek. Training connectionist models for the structured language model. In *Empirical Methods in Natural Language Processing (EMNLP'03)*, 2003.

Wei Xu and Alex Rudnicky. Can artificial neural networks learn language models? In *The 6th International Conference on Statistical Language Processing*, pages M1–13, Beijing, China, 2000.

Mikio Yamamoto and Kugatsu Sadamitsu. Dirichlet mixtures in text modeling. Technical Report CS Technical Report CS-TR-05-1, University of Tsukuba, 2005.

Sibel Yaman, Jen-Tzung Chien, and Chin-Hui Lee. Structural Bayesian language modeling and adaptation. In *Proc. of Interspeech'07*, pages 2365–2368, Antwerp, Belgium, August 2007.

Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.

Qiaohui Zhang, Atsumi, Imamiya, Kentaro Go, and Xiaoyang Mao. Designing a robust speech and gaze multimodal system for diverse users. In *Proceedings of IEEE International Conference on Information Reuse and Integration (IRI2003)*, pages 354–361, 27-29 October 2003.