HANDLING UNCERTAIN OBSERVATIONS IN UNSUPERVISED TOPIC-MIXTURE LANGUAGE MODEL ADAPTATION

Ekapol Chuangsuwanich[1], Shinji Watanabe[2§], Takaaki Hori[3], Tomoharu Iwata[3], James Glass[1]

[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, Massachusetts, USA

[2,3]NTT Communication Science Laboratories, NTT Corporation, Japan

[1]{ekapolc, glass}@mit.edu [2]watanabe@merl.com [3]{hori.t, iwata.tomoharu}@lab.ntt.co.jp

ABSTRACT

We propose an extension to the recent approaches in topic-mixture modeling such as Latent Dirichlet Allocation and Topic Tracking Model for the purpose of unsupervised adaptation in speech recognition. Instead of using the 1-best input given by the speech recognizer, the proposed model takes confusion network as an input to alleviate recognition errors. We incorporate a selection variable which helps reweight the recognition output, thus creating a more accurate latent topic estimate. Compared to adapting based on just one recognition hypothesis, the proposed model show WER improvements on two different tasks.

Index Terms— language model, latent topic model, topic tracking, confusion network.

1. INTRODUCTION

Adaptive and topic-mixture models have been explored by researchers in order to describe text corpora. Cache-based models exploit the property that words that appear earlier in the document are more likely to appear again [1]. Another popular approach is to model the underlying topic mixtures and interpolate between topic dependent word distributions [2]. One example of such an approach is Latent Dirichlet Allocation (LDA) which identifies topics from an unlabeled corpus in an unsupervised manner [3]. The topic mixtures can be used for document retrieval or document classification. These techniques can also be applied to help automatic speech recognition by adapting the Language Model (LM). For the task of speech recognition in academic lectures, Hsu and Glass [4] used a Hidden Markov Model with LDA (HMM-LDA) [5] which can model content words as well as syntactic words. In our previous work, we developed the Topic Tracking Language Model (TTLM) to explicitly capture the time evolution of topics throughout a recording session [6]. In [7], LDA is used with a class-based cache model to also incorporate topic history. However, the aforementioned approaches adapt the language model using the 1-best recognition results. Unlike text corpora where word observations are considered certain, ASR output is actually a set of uncertain observations with associated posterior probabilities. Despite error rates as high as 30-40% in some large vocabulary tasks, previous works typically use the most likely output from the speech recognizer. The improvement from the adaptation tends to diminish compared to the perplexity gains on text corpora, and sometimes it becomes even worse than the baseline.

In this work, we expand on our previous work by introducing a latent selection variable into existing methods such as TTLM and LDA to deal with confusion network inputs instead of the conventional bag-of-word inputs. The model then "selects" the word that best suits the current model parameters within a Gibbs sampling framework. Since topic-mixture models are capable of improving recognition results, incorporating latent topics should be able to reliably reweight the network. By using this model extension, we were able to improve the topic tracking capability and our ultimate recognition results in two different speech recognition tasks.

The rest of this paper is organized as follows. In the next section, we give a brief overview of TTLM which we will use as the basis to explain our proposed model. We then explain our extension of TTLM to cope with possible recognition errors. In Section 4, we explain our experiments and discuss the performance of our model. Finally, in Section 5 we provide some concluding remarks and describe some future plans.

2. TOPIC TRACKING LANGUAGE MODEL

In our previous work [6], we proposed TTLM which, unlike the original LDA, can capture the time evolution of topics. A long session of speech input is divided into chunks $t = 1, 2, ..., T$ that is modeled by different topic distributions $\boldsymbol{\phi}_t = \{\phi_{tk}\}_{k=1}^{K}$ where K is the number of topics. The current topic distribution depends on the topic distribution of the past H chunks and precision parameters α_{th} as follows:

$$P(\boldsymbol{\phi}_t|\{\hat{\boldsymbol{\phi}}_{t-h}, \alpha_{th}\}_{h=1}^{H}) \propto \prod_{k=1}^{K} \phi_{tk}^{(\alpha * \hat{\phi}_k)_t - 1} \quad (1)$$

where $(f * g)_t \triangleq \sum_{h=1}^{H} f_{th} g_{t-h}$ and $\hat{\phi}_{tk}$ is the mean of the

[§]Shinji Watanabe is now with Mitsubishi Electric Research Laboratories.

k^{th} topic distribution at chunk t.

TTLM can be applied on the 1-best hypothesis to recover the latent topic probability distributions. With the topic distribution, the unigram probability of a word w_m in the chunk can be recovered using the topic and word probabilities $P(w_m) \triangleq \sum_{k=1}^{K} \hat{\phi}_{tk}\theta_{kw_m}$, where θ_{kw_m} is the unigram probability of word w_m in topic k. The updated unigram can be used to scale n-grams via Minimum Discrimination Information (MDI) adaptation [8]. The adapted n-gram can be used for a 2^{nd} pass recognition for better results.

The original TTLM can also adapt the topic dependent unigrams. However, it was found that adapting the unigrams can cause degradations due to data sparsity when the chunk size is small. Due to this concern and for simplicity, we omit any discussions on unigram adaptation in this paper. The set of topic dependent unigrams, denoted by θ, is trained using LDA on the training corpus and kept fixed throughout.

3. TOPIC TRACKING LANGUAGE MODEL USING CONFUSION NETWORK INPUTS

Similar to TTLM, we start by breaking the speech input into chunks. However, we will use the resulting confusion network from the first pass recognition instead of the 1-best hypothesis. Consider a confusion network with M word slots. Each word slot m can contain different number of arcs A_m, with each arc containing a word w_{ma} and a corresponding arc posterior d_{ma}. As mentioned earlier, since the TTLM can help produce better transcriptions, there should be merit in using the latent topics discovered in the model to reweight the recognition outputs. Instead of fully trusting the 1-best hypothesis, the goal is to have the model be able to select the best arcs that can describe the chunks according to the latent topics. To accomplish this, we associate a binary selection parameter s_{ma}, where $s_{ma} = 1$ indicates that the arc is selected. S_m represents the arc index a where $s_{ma} = 1$. Selected arcs are considered to be correct words which should be generated from the current topic. The other arcs are considered errors generated from a separated error unigram θ_e. We denote W_t, Z_t, S_t as the sequence of words, topics, and selections in chunk t respectively, while the subscript m denotes the index within the chunk. A graphical representation of the Topic Tracking Language Model with Confusion Network inputs (TTLMCN) is show in Figure 1. The generative process of the TTLMCN can be summarized as follows:

1. Draw $\boldsymbol{\phi}_t$ from Dirichlet$((\alpha * \hat{\phi}_k)_t)$
2. For each word slot m in chunk t
   (a) Draw S_m from Multinomial(D_{tm})
   (b) Draw z_m from Multinomial$(\boldsymbol{\phi}_t)$
   (c) For each arc a in word slot m
   - If $s_{ma} = 1$, draw w_{ma} from Multinomial(θ_{tz_m})
   - If $s_{ma} = 0$, draw w_{ma} from Multinomial(θ_e)

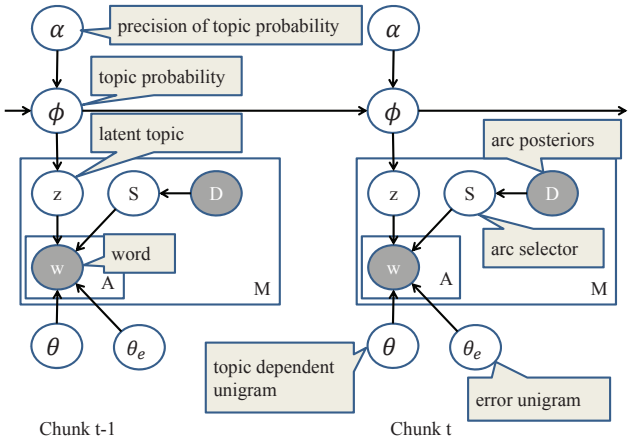

Fig. 1. Graphical representation of TTLMCN. Shaded and unshaded nodes indicate observed and latent variables, respectively. Note that ϕ can depend on the ϕs of several preceding chunks. However, for figure clarity, we depict only the case when $H = 1$.

From this generation process, for each chunk t, we can write the joint distribution of words, latent topics and arc selections conditioned on the topic probabilities, unigram probabilities, and arc posteriors as follows:

$$P(W_t, Z_t, S_t|\theta_e, \theta, \boldsymbol{\phi}_t, D_t) = \prod_{m}^{N} \phi_{tz_m} \prod_{a}^{A_m} d_{ma}^{s_{ma}} \theta_{z_m w_{ma}}^{s_{ma}} \theta_{ew_{ma}}^{1-s_{ma}} \quad (2)$$

3.1. Inference

We infer latent topics and "correct" words based on collapsed Gibbs sampling [9]. We start from the joint distribution and substituting Eqs. 1 and 2 :

$$\begin{aligned} &P(W_t, Z_t, S_t|\hat{\boldsymbol{\phi}}_{t-1}, D_t, \alpha_t, \theta, \theta_e) \\ &= \int P(W_t, Z_t, S_t|\theta_e, \theta, \boldsymbol{\phi}_t, D_t) P(\boldsymbol{\phi}_t|\hat{\boldsymbol{\phi}}_{t-1}, \alpha_t) d\phi_t \\ &= \frac{\Gamma(\alpha_t)}{\prod_k \Gamma((\alpha * \hat{\phi}_k)_t)} \frac{\prod_k \Gamma(n_{tk} + (\alpha * \hat{\phi}_k)_t)}{\Gamma(n_t + \sum_{h=1}^{H} \alpha_{th})} \\ &\times \prod_m \prod_a d_{ma}^{s_{ma}} \theta_{z_m w_{ma}}^{s_{ma}} \theta_{ew_{ma}}^{1-s_{ma}} \end{aligned} \quad (3)$$

where $\Gamma()$ is the Gamma function, n_t denotes the number of word slots in chunk t, and n_{tk} denotes the number of words assigned to topic k in chunk t.

From Eq. 3, we can derive the Gibbs sampling equations of Z_t and S_t. (Appendixes of [6] can be used as an outline for the derivations.)

$$\begin{aligned} P_{tmkj}^{(z)} &\triangleq P(z_m = k|S_m = j, ...) \\ &\propto \frac{n_{tk\backslash m} + (\alpha * \hat{\phi}_k)_t}{n_{t\backslash m} + \sum_{h=1}^{H} \alpha_{th}} \theta_{kw_{mj}} \end{aligned} \quad (4)$$

$$P_{tmkj}^{(s)} \triangleq P(S_m = j|z_m = k, ...) \propto d_{mj}\theta_{kw_{mj}} \quad (5)$$

where $\backslash m$ implies counts excluding the m^{th} word slot. These two equations make sense intuitively: the probability of picking a topic depends on how often the topic occurs (with some influence from previous chunks). The probability of selecting an arc depends on the posterior and the corresponding topic dependent unigram.

α_{th} can be updated by the maximum likelihood estimation of the joint distribution in Eq. 3 as follows:

$$\alpha_{th} \leftarrow \alpha_{th} \frac{\sum_k \hat{\phi}_{t-hk}(\Psi(n_{tk} + (\alpha * \hat{\phi}_k)_t) - \Psi((\alpha * \hat{\phi}_k)_t))}{\Psi(n_t + \sum_{h'} \alpha_{th'}) - \Psi(\sum_{h'} \alpha_{th'})} \tag{6}$$

where $\Psi()$ is the Digamma function. By iterating Eqs. (4)-(6), we can obtain Z_t and α_{th} which can then be used to estimate the means of the topic distribution as follows:

$$\hat{\phi}_{tk} = \frac{n_{tk} + (\alpha * \hat{\phi}_k)_t}{n_t + \sum_{h'} \alpha_{th'}} \tag{7}$$

These means can be used to update the n-gram just like in the original TTLM. Extending to LDA is very similar in concept. One thing of note is the handling of epsilon transitions which indicates a deletion of the word slot. However, unigrams trained on text corpora would not naturally include epsilon transitions. We incorporate this by adding an additional entry to the unigrams with probability p_ϵ and rescale the rest of the probability masses accordingly. p_ϵ is considered a tuning parameter indicating the trade-off between insertions and deletions. The word counts should also exclude word slots that select epsilon transition arcs.

3.2. Posterior Interpolation

Even though arc selection based on latent topics makes intuitive sense, basing these selections on unigrams instead of n-grams causes many incorrect arcs to be selected. One possible fix is to have a high pruning threshold for generating the confusion network. This leaves only the most confusable choices which helps reduce selection errors. However, the correct arcs might also be pruned out, limiting the usefulness of this model. Thus, we use a simple interpolation scheme to reinforce the original posterior by modifying the Gibbs sampling of the arc selection in Eq. 5 as follows:

$$P'(S_m = j | z_m = k, ...) = P^{(z)}_{tmkj} P^{(s)}_{tmkj} + (1 - P^{(z)}_{tmkj}) d_{mj} \tag{8}$$

4. EXPERIMENTS

We conducted experiments on two different speech recognition tasks; the MIT OpenCourseWare (MIT-OCW) [10] and the Corpus of Spontaneuos Japanese (CSJ) [11]. The TTLM implementation followed the framework described in [6]. Each recording was divided into chunks that were then processed sequentially. The error unigram θ_e was set to be uniform. For both tasks, we show the performance of the existing baselines: unadapted LM (BASE), LDA, and TTLM. We also show the performance of LDA and TTLM with our proposed extensions to accept confusion network inputs (LDACN and TTLMCN). The posterior interpolation method was also examined for the TTLMCN case (TTLMCNI). For the MIT-OCW task, we also conducted an oracle experiment (ORACLE) where the means of the latent topic were learned from the original transcripts to indicate the best case scenario for language model adaptation based on TTLM.

System	P1	P3	WER	% change
BASE	611.5	208.2	41.4	-
LDA	543.2	187.8	41.3	0.2
LDACN	549.2	202.4	41.0	1.0
TTLM	521.7	184.9	41.0	1.0
TTLMCN	504.4	179.4	40.6	2.0
TTLMCNI	503.4	178.88	40.5	2.2
ORACLE	482.9	171.9	39.4	4.8

Table 1. Test set performance on the MIT-OCW task. P1 and P3 denote 1-gram and 3-gram perplexity respectively.

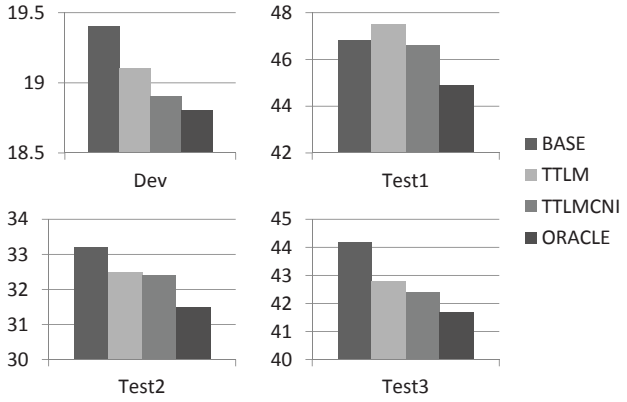

Fig. 2. WER performance on each individual lecture.

4.1. MIT-OpenCourseWare Corpus

MIT-OCW is mainly composed of lectures given at MIT. Each lecture is typically two hours long. We segmented the lectures using Voice Activity Detectors into utterances averaging two seconds each. In order to be consistent with our previous work in [6], we set the size of each chunk to 64 utterances, although our more recent work [12] incorporates multiple chunk sizes for additional performance gain. The training data consists of 147 lectures (128h of speech, 6.2M word). The development set contains one lecture (1 hour), while the evaluation was done on three lectures (3.5 hours).

Table 1 summarizes the recognition results on this task. Incorporating confusion network inputs improved performance for both LDA and TTLM algorithms. The posterior interpolation also improved the performance slightly. Figure

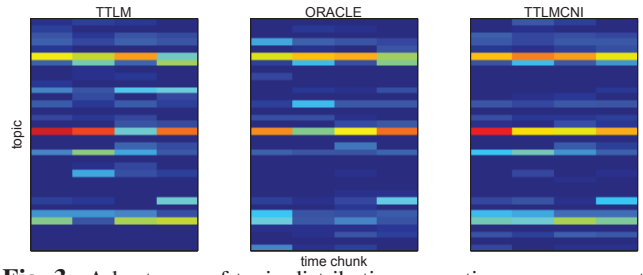

Fig. 3. A heat map of topic distributions over time on an excerpt of TEST3. Blue pixels represent lower probability, while red pixels represent higher probability. Y-axis corresponds to different LDA topics. X-axis corresponds to 4 successive time chunks.

System	Dev WER	Test WER	% change
BASE	17.7	20.8	-
LDA	16.5	19.8	4.8
LDACN	16.4	19.6	5.8
TTLM	16.3	19.7	5.2
TTLMCN	16.1	19.5	6.5
TTLMCNI	16.1	19.4	6.7

Table 2. Performance on the CSJ task.

2 shows the performance of each algorithm for each individual lecture. One important thing to note is that on Test1, TTLM actually did worse than the non-adapted LM. This was partly due to the high WER of the ASR. Another cause was that the chunks in Test1 typically contained fewer words than the other lectures, which intensified the WER problem. Incorporating confusion network inputs alleviated the problem and improved the recognition results compared to the baseline. However, there was still a sizable gap from the oracle experiments indicating room for possible improvements. We also looked into the difference between the topic probability distributions estimated by each model such as one shown in Figure 3. We can see that the topic probability of TTLMCNI is more similar to the oracle experiment than TTLM, especially in the low probability regions. The average KL divergence between the distributions obtained in TTLM and ORACLE was 3.3, while the KL divergence between TTLMCNI and ORACLE was 1.3, a noticeable improvement.

4.2. Corpus of Spontaneous Japanese

CSJ is mainly composed of conference presentations. The acoustic model was trained on 234 hours of speech, while the LM was trained on a larger set of 6.8M words. We used "CSJ testset 2" for development and "CSJ testset 1" for testing. Each set contains ten presentations where each presentation averages around 15 minutes. We used one utterance as one chunk to simulate a more real-time friendly scenario. Unlike in our previous work [6] where the Minimum Discrimination Information n-gram scaling factor was fixed at 0.5, we also tuned this parameter using the development set.

Table 2 shows the performance on the CSJ task. The confusion network extensions continued to show similar improvements even though the baseline WER was already low compared to the MIT-OCW task. The TTLMCNI showed the best improvement of 6.7% compared to the baseline LM.

5. CONCLUSION

We described an extension for the TTLM in order to handle errors in speech recognition. The proposed model used a confusion network as input instead of just one ASR hypothesis which improved performance even in high WER situations. Experiments on MIT-OCW and CSJ tasks showed improvements on the WER. The gain in word error rate was not very large since the LM typically contributed little to the performance of LVCSR. However, the topic probability estimates improved considerably. As future work, we would like to explore other ways of reincorporating the n-gram into the selection which might decrease the gap in performance between our model and the oracle.

6. REFERENCES

[1] R. Kuhn and R. De Mori, "A cache-based natural language model for speech recognition.," *IEEE Trans. Pattern Analysis and Machine Intelligence.*, vol. 6, no. 12, pp. 570–583, 1990.

[2] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. UAI*, 1999, pp. 289–296.

[3] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, , no. 3, pp. 993–1022, 2003.

[4] B.J. Hsu and J. Glass, "Style & topic language model adaptation using HMM-LDA.," in *Proc. EMNLP*, 2006, pp. 373–381.

[5] T. Griffiths, M. Steyvers, and J. Tenenbaum, "Integrating topics and syntax.," in *Advance in Neural Information Processing Systems*, 2004, vol. 17, pp. 537–544.

[6] S. Watanabe, T. Iwata, et al., "Topic tracking language model for speech recognition," *Computer Speech and Language*, vol. 25, pp. 440 –461, 2011.

[7] J.T. Chien and C.H. Chueh, "Dirichlet class language models for speech recognition," in *IEEE trans. on Audio, Speech, and Language processing*, 2011, vol. 19, pp. 482–495.

[8] R. Kneser, J. Peters, and D. Klakow, "Language model adaptation using dynamic marginals.," in *Proc. Eurospeech*, 1997, pp. 1971–1974.

[9] T. Griffiths and M. Steyvers, "Finding scientific topics.," in *Proc. of the National Academy of Sciences*, 2004, vol. 101, pp. 5228–5235.

[10] J. Glass, T.J. Hazen, et al., "Recent progress in the MIT spoken lecture processing project.," in *Proc. Interspeech*, 2007, pp. 2553–2556.

[11] S. Furui, K. Maekawa, and M. Isahara, "A Japanese national project on spontaneous speech corpus and processing technology.," in *Proc. ASR*, 2000, pp. 244–248.

[12] S. Watanabe, A. Nakamura, and Juang B.H., "Model adaptation for automatic speech recognition based on multiple time scale evolution.," in *Proc. Interspeech*, 2011, pp. 1081–1084.