AN HMM-BASED TEXT SEGMENTATION METHOD USING VARIATIONAL BAYES APPROACH AND ITS APPLICATION TO LVCSR FOR BROADCAST NEWS

Takafumi Koshinaka, Ken-ichi Iso, and Akitoshi Okumura

Media and Information Research Laboratories, NEC Corporation, Kawasaki, JAPAN
koshinak@ap.jp.nec.com

ABSTRACT

Recent progress in large vocabulary continuous speech recognition (LVCSR) has raised the possibility of applying information retrieval techniques to the resulting text. This paper presents a novel unsupervised text segmentation method. Assuming a generative model of a text stream as a left-to-right hidden Markov model (HMM), text segmentation can be formulated as model parameter estimation and model selection using the text stream. The formulation is derived based on the variational Bayes framework, which is expected to work well with highly sparse data such as text. The effectiveness of the proposed method is demonstrated through series of experiments, where broadcast news programs are automatically transcribed and segmented into separate news stories.

1. INTRODUCTION

Recent progress in large vocabulary continuous speech recognition (LVCSR) has raised the possibility of applying information retrieval techniques to the resulting text. Text segmentation is one of such techniques. The goal of a text segmenter is to segment a text stream into some semantically cohesive units, namely, stories. If there are large-scale audio-video archives such as a large number of broadcast news programs, and if an LVCSR system is available, the text segmenter can achieve audio-video segmentation [1], which is useful meta data in information retrieval.

Most conventional methods for text segmentation are based on the "change point detection" (CPD) approach, a typical implementation of which is Hearst's TextTile [2]. TextTile regards a text stream as a word sequence, and sets a constant-width sliding window along the sequence. The sliding window has its own word frequency vector at each location, and computes the similarity between adjacent windows (usually defined as the cosine between word frequency vectors). Thus, story boundaries, namely the points when topics change, can be detected by observing local minimum points on the similarity measure.

The CPD approach contains some parameters, and their values must be appropriately tuned by skilled engineers in order to achieve good performance. For example, the width of the sliding window depends on the story lengths, so that the width is usually decided by assuming some distribution of story lengths. In many cases, such an assumption decreases the flexibility of a system. That is to say, the system may return unexpected output when input data is outside the assumption.

In this paper, we propose a parameter-free text segmentation method, which assumes a left-to-right hidden Markov model (HMM) as a generative model of a text stream. Each state of the HMM is associated with a topic, and generates a story related to the topic. Then, text segmentation can be treated as a process of fitting HMMs to the input text stream. In other words, text segmentation can be formulated as HMM learning using the input text stream. Furthermore, we introduce Bayes estimation, namely, variational Bayes [3], into HMM learning. We experimentally show the effectiveness of the proposed method, which is expected to have high robustness against sparse data such as text streams.

Note that there is an existing HMM-based text segmentation method by Yamron et al [1]. They employed an ergodic HMM fully trained with prepared large-scale corpora. That is, they treated segmentation as decoding of the input text stream using the HMM. In contrast, our approach is unsupervised. We employ a left-to-right HMM, which is initialized and trained with each input text stream only.

2. TEXT SEGMENTATION BY MODEL FITTING

2.1. Basic Concept

We assume that a text stream of N topics is generated from a discrete HMM of N-state left-to-right architecture (Figure 1). The HMM starts at the initial state $I = 0$, and repeats state transitions according to the transition probabilities a_i ($i = 1, \cdots, N$) until it reaches the final state $F = N+1$. At each state transition, the HMM outputs a word according to the discrete output probability b_{ij} ($j = 1, \cdots, L$) associated with state i. Once the HMM reaches the final state, you have observed a word sequence $o_1, \cdots, o_T$ with N topics and lexicon size L.

Text segmentation can be interpreted as an inverse operation to the text generation described above. A given word sequence $O = (o_1, \cdots, o_T)$ can be segmented into N subsequences of words by fitting an N-state HMM to the text stream. This fitting operation is formulated as model parameter estimation from the observation O.

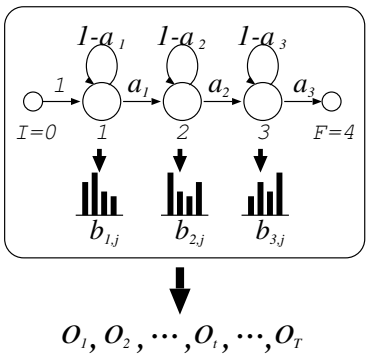

Fig. 1. A generative model of a text stream containing three stories. Each state parameterizes a topic as b_{ij}.

2.2. Maximum Likelihood Segmentation

Let us consider ML estimation of an HMM parameterized by $\theta = \{a_i, b_{ij} \mid i = 1, \cdots, N, \quad j = 1, \cdots, L\}$ using word sequence $O = (o_1, \cdots, o_T)$. This can be easily realized by applying the expectation maximization (EM) algorithm, namely, the Baum-Welch parameter reestimation formulae. As is well known, the Baum-Welch uses auxiliary variables known by the name of forward and backward variables $\alpha_t(i)$ and $\beta_t(i)$ respectively. They are defined as follows:

$$\begin{cases} \alpha_t(i) = P(z_{t,i} = 1, o_1, \cdots, o_t \mid \theta) \\ \beta_t(i) = P(o_{t+1}, \cdots, o_T \mid z_{t,i} = 1, \theta) \end{cases}, \qquad (1)$$

where $z_{t,i}$ is a binary latent variable that takes 1 if the HMM stays at state i after the tth state transition (if the tth word observation o_t has come out from state i), otherwise it takes 0. According to Eqs.(1), you obtain the following probability that the word o_t has come from state i:

$$P(z_{t,i} = 1 \mid O, \theta) = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\,\beta_t(j)}. \qquad (2)$$

Eq.(2) designates a *soft* clustering of words $o_1, \cdots, o_T$ into N clusters, and each cluster is actually a connected segment. Making an N-state left-to-right HMM learn text stream O is equivalent to finding a sequence of N homogeneous segments in O, where "homogeneous" means constancy of word frequency. Therefore, once the learning process has fully converged, each segment obtained by computing Eq.(2) is expected to be a semantically cohesive segment containing one topic in itself.

2.3. Bayesian Segmentation

It is also possible to apply Bayes estimation to our text segmentation method. The objective of Bayes estimation is to obtain parameter distribution $p(\theta \mid O)$. In contrast, ML estimation is a kind of point estimation. The distribution contains information about whether the amount of training data O is sufficient or not. Hence the Bayes estimation is expected to work well with sparse data such as a text stream.

Although Bayes estimation is supposed to be computationally difficult for stochastic models with latent variables, an effective algorithm called variational Bayes (VB) has been available in recent years [3]. We introduce the VB algorithm into our segmentation task. Here, we present the final result of the VB formulation for HMMs.

1. Assume the following prior distribution with respect to $\theta = \{a_i, b_{ij}\}$:

$$p(\theta) = \prod_{i=1}^{N} \mathcal{B}(a_i \mid \kappa_{1,i}, \kappa_{0,i}) \times \prod_{i=1}^{N} \mathcal{D}(b_{i,1}, \cdots, b_{i,L} \mid \lambda_{i,1}, \cdots, \lambda_{i,L}),$$

where $\mathcal{B}(x \mid a, b) \propto x^{a-1}(1-x)^{b-1}$ and $\mathcal{D}(x_1, \cdots, x_n \mid \phi_1, \cdots, \phi_n) \propto x_1^{\phi_1 - 1} \cdots x_n^{\phi_n - 1}$ denote the beta and Dirichlet distribution respectively. Here, $\kappa_{0,i}$, $\kappa_{1,i}$, λ_{ij} are called hyper-parameters, which control parameter distribution. Now we employ notation $\kappa_{0,i}$, $\kappa_{1,i}$, λ_{ij} for hyper parameters for the prior, and $\kappa_{0,i}^{(l)}$, $\kappa_{1,i}^{(l)}$, $\lambda_{ij}^{(l)}$ for the posterior, where l denotes the number of VB iterations.

2. Calculate (Bayesian) forward and backward variables using the following recurrence formulae:

$$\alpha_1(i) = \exp(B_{i,o_1})\,\delta_{i,1}, \quad \beta_T(i) = \exp(A_{1,N})\,\delta_{i,N},$$

$$\begin{aligned} \alpha_{t+1}(i) &= \alpha_t(i-1)\exp\left(A_{1,i-1} + B_{i,o_{t+1}}\right) \\ &+ \alpha_t(i)\exp\left(A_{0,i} + B_{i,o_{t+1}}\right), \end{aligned}$$

$$\begin{aligned} \beta_{t-1}(i) &= \beta_t(i)\exp\left(A_{0,i} + B_{i,o_t}\right) \\ &+ \beta_t(i+1)\exp\left(A_{1,i} + B_{i+1,o_t}\right), \end{aligned}$$

where

$$A_{j,i} = \Psi\left(\kappa_{j,i}^{(l)}\right) - \Psi\left(\kappa_{0,i}^{(l)} + \kappa_{1,i}^{(l)}\right) \quad (j = 0, 1),$$

$$B_{ik} = \Psi\left(\lambda_{ik}^{(l)}\right) - \Psi\left(\sum_{j=1}^{M} \lambda_{ij}^{(l)}\right),$$

and $\alpha_{t,0} = \beta_{t,N+1} = 0$ $(t = 1, \cdots, T)$. Here, δ_{ij} denotes Kronecker's delta, and $\Psi(x)$ denotes the digamma function defined as $\Psi(x) = \Gamma'(x)/\Gamma(x) = (\log\Gamma(x))'$.

3. Calculate the expectations of the latent variables $z_{t,i}$ as follows:

$$\overline{z_{t,i}} = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\,\beta_t(j)}, \tag{3}$$

$$\overline{z_{t,i}z_{t+1,i}} = \frac{\alpha_t(i)\exp\left(A_{0,i}+B_{i,o_{t+1}}\right)\beta_{t+1}(i)}{\sum_{j=1}^{N}\alpha_t(j)\,\beta_t(j)},$$

$$\overline{z_{t,i}z_{t+1,i+1}} = \frac{\alpha_t(i)\exp\left(A_{1,i}+B_{i+1,o_{t+1}}\right)\beta_{t+1}(i+1)}{\sum_{j=1}^{N}\alpha_t(j)\,\beta_t(j)}$$

4. Update the hyper-parameters as follows:

$$\kappa_{0,i}^{(l+1)} = \kappa_{0,i} + \sum_{t=1}^{T-1}\overline{z_{t,i}z_{t+1,i}},$$

$$\kappa_{1,i}^{(l+1)} = \kappa_{1,i} + \sum_{t=1}^{T-1}\overline{z_{t,i}z_{t+1,i+1}} + \delta_{N,i},$$

$$\lambda_{ik}^{(l+1)} = \lambda_{ik} + \sum_{t=1}^{T}\delta_{k,o_t}\overline{z_{t,i}}.$$

5. Go back to Step 2 until convergence.

Once the estimation process above has converged, the segmentation result can be obtained in the same way as shown in the previous subsection using Eq.(2), which corresponds to Eq.(3) in Bayesian segmentation. Note that the left hand side of Eq.(3) is an expectation of that of Eq.(2) with respect to θ, that is to say, $\overline{z_{t,i}} = \int P(z_{t,i}=1 \mid O,\theta)\,p(\theta \mid O)\,d\theta$.

2.4. Number of Topics and Model Selection

Because it is generally unknown how many topics (segments) are contained in a text stream, any text segmenter must provide a means of determining the number of topics. In the proposed method, the number of topics corresponds to the number N of states in the HMM, so that model selection methods are applicable to our problem. You can determine the number of topics by preparing multiple hypotheses on N such as $N_{\min} \le N \le N_{\max}$, performing HMM learning for each hypothesis and choosing the best hypothesis based on a particular model selection criterion.

Several criteria for model selection exist. In ML estimation, you can choose Akaike's information criterion (AIC) or Rissanen's minimum description length (MDL) criterion as the model selection criterion. In Bayes estimation, the VB framework includes a model selection criterion, namely, a model posterior distribution $p(N \mid O) \propto p(O \mid N)\,p(N)$, which allows you to choose the most likely number of topics in the sense of maximum a posteriori probability (MAP).

3. EXPERIMENTS

Now we experimentally show the effectiveness of our text segmentation method. First, we chose a 15-minute Japanese TV broadcast news program, and randomly collected five different broadcasts (75 minutes) of this program. They consisted of 66 short news stories in total.

Then we transcribed them to prepare two sets of text data, one of which was manually transcribed, and the other automatically transcribed by our LVCSR system. We preliminarily calculated the word error rate (WER) of the LVCSR system for each broadcast and found that WER ranged from 11.6 to 19.5% and that the overall average was 15.5%. We also investigated the manually transcribed text data set, and found that the lengths of the news stories ranged from 16 to 699 words and averaged 180 words.

We defined a set of stopwords that included pronouns, conjunctions, auxiliary verbs, interjections, postpositional particles and so forth. Any words included in the set were removed from the text data sets before segmentation. After this preprocessing, the average story length decreased to 91 words, approximately half of the original length.

In our experiments, the goal of text segmentation was to segment a sequence of news stories into each separate story. We used a co-occurrence agreement probability [4] (CoAP) to measure segmentation accuracy. CoAP is a highly flexible measure, and a simplified version is widely used. We also adopted the simplified version, which is defined as the rate of correctly labeled word pairs within all pairs that are a fixed distance (k words) apart, where k is chosen to be half the average reference story length.

First, we performed a segmentation experiment on the assumption that the true number N of topics was known, where we segmented an N-topic text stream using an N-state HMM and not using any other HMMs. For test data, we extracted all possible sub-sequences of N stories (topics) from the text data set, where $5 \le N \le 10$.

Figure 2 plots the average segmentation accuracies by ML and Bayes segmentation to manually and automatically transcribed text streams for each N. This result shows that Bayes segmentation performs much better than ML segmentation. ML estimation potentially assumes the infinite data amount and is therefore too sensitive to words that occur with low frequency, while Bayes estimation is able to take into account data insufficiency, which text streams ordinarily imply. That is why we rate the Bayes segmentation performance higher.

On the other hand, there seems to be a trend for accuracy to decrease slightly as the number of topics increases in both EM and Bayes segmentation. We assume this is caused by a local maximum problem in training HMMs. Actually, we inspected text data with low segmentation accuracy, and found that accuracy sometimes depends on the

initial training condition. Hence there is a possibility that we can achieve further improvement by replacing the initialization process with something other than a simple "flat start".

Comparing manual and automatic (LVCSR) transcription, the accuracy of the latter is lower than that of the former. This is quite reasonable because LVCSR transcription contains a 15% recognition error rate on average. However, this level of accuracy is not that bad when compared to conventional methods, as will be shown later.

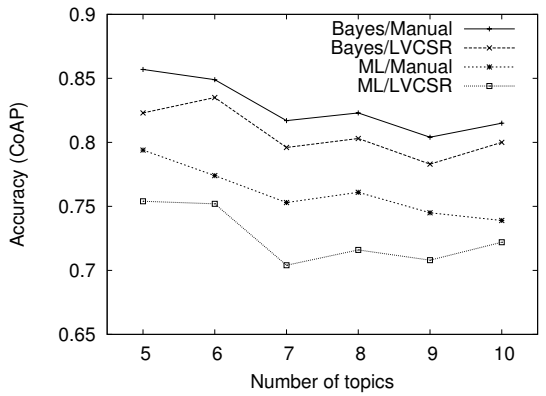

Fig. 2. Segmentation accuracies of the proposed ML and Bayes segmentation method to manually (Manual) and automatically (LVCSR) transcribed text streams.

Secondly, we performed another experiment under a more realistic condition where the true number N of topics was unknown. We also compared the proposed method with one of the conventional methods, namely, TextTile. Test data was exactly the same as in the previous experiment.

In the proposed method we applied Bayes segmentation, which showed better performance in the previous experiment. Consequently, the number of topics was determined by the Bayesian model posterior distribution, where the lower and upper bounds were set to $N_{\min} = 3$ and $N_{\max} = 12$.

In the conventional method, there were at least three tunable parameters: (1) window width, (2) the number of iterations in smoothing the cosine similarity and (3) the lower bound of a story length. Here we assumed (3) was identical to (1), and optimized (1) and (2). Note that you should be careful so as to conduct *open-data* experiments. In this case, we used a leave-one-out procedure, that is to say, leaving one broadcast out for testing, and optimizing parameters (1) and (2) using the remaining four broadcasts. We repeated this small experiment five times by rotating the test data.

The result shows that the proposed method produces higher accuracy in both manually and automatically transcribed text streams (Table 1). And furthermore, the proposed method works on automatically transcribed (error-contained) text streams with small degradation of accuracy compared to the conventional method.

Table 1. Mean segmentation accuracies of the proposed method and the conventional method to manually (Manual) and automatically (LVCSR) transcribed text streams.

	Conventional	**Proposed**
Manual	0.785	**0.845**
LVCSR	0.749	**0.829**

4. CONCLUSION

We proposed a novel parameter-free text segmentation method, which was formulated as (variational) Bayes estimation of an HMM from an input text stream. We experimentally showed that the proposed method achieved good performance in segmenting text streams transcribed from broadcast news programs using LVCSR.

Our future work includes reducing the local maxima arising in long text streams, and introducing tied-mixture architecture to HMMs. A tied-mixture HMM can be described with fewer parameters and with high scalability. It will reduce data sparseness occurring in text streams. Furthermore, such a model is mathematically interesting as dynamic generalization of the probabilistic latent semantic analysis (PLSA) [5].

5. REFERENCES

[1] J. Yamron, I. Carp, L. Gillick, S. Lowe, and P. Van Mulbregt, "Hidden markov model approach to text segmentation and event tracking," *Proc. ICASSP98*, 1998.

[2] M. Hearst, "Multi-paragraph segmentation of expository text," *32nd. Annual Meeting of the Association for Computational Linguistics*, 1994.

[3] H. Attias, "Inferring parameters and structure of latent variable models by variational bayes," *Proc. 15th Conf. on Uncertainty in Artificial Intelligence*, 1999.

[4] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," *Machine Learning, Vol.34, No.1-3, pp.177–210*, 1999.

[5] T. Hofmann, "Probabilistic latent semantic indexing," *Proc. 22nd Int'l Conf. on R&D in Information Retrieval (SIGIR'99)*, 1999.