Probabilistic Latent Semantic Analysis for Broadcast News Story Segmentation

Mimi Lu[1,2], *Cheung-Chi Leung*[2], *Lei Xie*[1], *Bin Ma*[2], *Haizhou Li*[2]

[1]Shaanxi Provincial Key Laboratory of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, China
[2] Institute for Infocomm Research, A*STAR, Singapore

mlu@nwpu-aslp.org lxie@nwpu.edu.cn {ccleung,mabin,hli}@i2r.a-star.edu.sg

Abstract

This paper proposes to perform probabilistic latent semantic analysis (PLSA) for broadcast news (BN) story segmentation. PLSA exploits a deeper underlying relation among terms beyond their occurrences thus conceptual matching can be employed to replace literal term matching. Different from text segmentation, lexical based BN story segmentation has to be carried out over LVCSR transcripts, where the incorrect recognition of out-of-vocabulary words inevitably impacts the semantic relation. We use phoneme subwords as the basic term units to address this problem. We integrate a cross entropy measurement with PLSA to depict lexical cohesion and compare its performance with the widely used cosine similarity metric. Furthermore, we evaluate two approaches, namely TextTiling and dynamic programming (DP), for story boundary identification. Experimental results show that the PLSA based methods bring a significant performance boost to story segmentation and the cross entropy based DP approach provides the best performance.

Index Terms: story segmentation, probabilistic latent semantic analysis, cross entropy, dynamic programming, spoken document retrieval

1. Introduction

Story segmentation refers to the task of partitioning a stream of text, speech or video into continuous units, each addressing a main topic. It serves as a necessary precursor to various tasks, such as topic detection and tracking, information retrieval and summarization, etc. Specifically, for broadcast news retrieval, it is preferred that the short clip related to the user's exact interests is returned by the retrieval system rather than the entire program. Manual segmentation is labor-intensive and infeasible due to the exponential growth of multimedia data. Thus automatic story segmentation is highly in demand.

For story segmentation, three categories of cues, including lexical, acoustic and visual features, are typically exploited. While visual and acoustic cues rely heavily on editorial rules, lexical cues are more generic because they reveal topic shifts via semantic variations in text. Words in a topic usually agglomerate via inter-word semantic relations and different topics tend to deploy different word usage. This phenomenon is known as lexical cohesion. TextTiling [1] is a typical lexical cohesion based segmentation technique. It measures pairwise sentence lexical similarities in a text, and identifies boundaries at local similarity minima. This is one of the methods focusing on identifying boundaries through local comparison, while other lexical similarity based methods applying dynamic programming (DP) algorithm [2, 3], aim at finding an optimal segmentation under some global criteria. Comparisons in [3] showed that DP offers better performances.

The lexical cohesion based approaches mentioned above mostly rely on rigid word repetition. Yet it is well known that this *literal term matching* has several drawbacks: First of all, there are many ways to express a certain concept and thus seeking relevant texts simply through strict word comparison may fail. Secondly, a word may have multiple senses and manifold types of usage and hence similarity on word occurrence may convey weak conceptual homogeneity. In a word, individual terms provide unreliable evidence about their concepts. Therefore, several strategies which take *conceptual matching* into account are introduced. These methods attempt to explore some underlying latent semantic structure in the data, which is partially obscured by the randomness of word choices. Latent semantic analysis (LSA) based story segmentation [4] employs the contextual meaning of word usage and improves separability among different topics over conventional lexical approaches. However, its methodological foundation remains unsound and it is insufficient to explicitly capture multiple senses of a word [5]. As a probabilistic variant of LSA, probabilistic latent semantic analysis (PLSA) has a solid statistical foundation and defines a proper generative data model that has been proven to provide better performance than LSA.

PLSA based text segmentation has been well-studied [6], but using the same approach for spoken document segmentation should take further consideration. Segmentation of spoken documents performs on erroneous words from a large vocabulary continuous speech recognizer (LVCSR) output. Speech recognition errors induce noises on words and break lexical cohesion, which result in both term and conceptual matching failures. Out-of-vocabulary (OOV) words constitute a large part of the recognition errors. However these OOV words cannot be neglected as they are typically name entities that are key to topics and their mis-recognition remains the major obstacle for broadcast news segmentation. In contrast to word recognition, phoneme recognition has the advantage of partial matching since the incorrectly recognized words may contain subword units correctly recognized [7].

In this paper, we apply PLSA to story segmentation for broadcast news. We propose to use phoneme *n*-gram as the basic term unit to measure lexical cohesion for overcoming OOV problem. Moreover, a cross entropy based measurement is introduced to depict lexical distance. Cross entropy is a divergence measurement based on information theory, which is used to describe the dissimilarity between two probability distributions [8]. It has been successfully adopted as a classifier in text classification task [9], a ranking function for information retrieval [10] and etc. When PLSA is adopted, we compare the performance of using cross entropy and the widely used cosine similarity for lexical cohesion measure. Furthermore, the performances of using TextTiling and DP for story boundary identification are compared.

2. Probabilistic latent semantic analysis

Probabilistic Latent Semantic Analysis (PLSA) was first introduced in information retrieval [11]. In a PLSA model, each co-occurrence observation, i.e., the occurrence of a word $w \in \mathcal{W}$ in a particular document $d \in \mathcal{D}$, is associated with an unobserved variable $z \in \mathcal{Z}$, which can be considered as a class label or topic. Given the assumption that d and w are independently conditioned on the state of the associated latent variable z, a joint probability model of document d and word w can be defined by:

$$P(d,w) = P(d)P(w|d), P(w|d) = \sum_{z \in Z} P(w|z)P(z|d) \quad (1)$$

The probabilities $P(w|z)$ and $P(z|d)$ are two parameters to be learnt in the PLSA model. An iterative Expectation Maximization (EM) algorithm is adopted for the maximum likelihood estimation by maximizing:

$$L = \sum_{d \in D} \sum_{w \in W} n(d,w) \log P(d,w) \quad (2)$$

where $n(d,w)$ denotes the frequency of word w in document d. Starting from random initial values, EM procedure alternates two steps: i) E-step where posterior probabilities of latent variables given the observations are computed based on the current estimates of model parameters as:

$$P(z|d,w) = \frac{P(w|z)P(z|d)}{\sum_{z'} P(w|z')P(z'|d)} \quad (3)$$

and ii) M-step, where Eq.(2) is maximized by re-estimating parameters $P(w|z)$ and $P(z|d)$ with the new expected values $P(z|d,w)$ as:

$$P(w|z) = \frac{\sum_d n(d,w)P(z|d,w)}{\sum_{w'} \sum_d n(d,w')P(z|d,w)}, \quad (4)$$

$$P(z|d) = \frac{\sum_w n(d,w)P(z|d,w)}{\sum_{z'} \sum_w n(d,w)P(z'|d,w)} \quad (5)$$

After learning parameters over documents from the training corpus, the estimated $P(w|z)$ are used to compute $P(z|q)$ for unseen documents q through a *folding-in* process [11]. The process consists of maximizing the likelihood of the new document q with a partial version of EM algorithm described above: the E-step is identical while in the M-step $P(w|z)$ are kept fixed and only $P(z|q)$ are updated.

3. Data preparation

In the preparation stage, all texts in the corpus are preprocessed by i) tokenization, ii) units formation, and iii) vectorization. Tokenization process also involves stop-word removal and stemming.

In the units formation step, the data for PLSA model training and for story segmentation evaluation are treated in slightly different manners. For the training collection, LVCSR transcripts with manually labeled boundary tags are used. Text streams are broken into non-overlapping block units, where each block is actually a real story. For documents to be segmented, since story boundaries are assumed to appear at sentence boundaries and there are no boundary information of real sentences in the transcripts, we divide texts to fixed-size blocks as the elementary units and sliding windows are adopted over these blocks.

In the vectorization step, each block *b* is represented by a vector consisting of term counts in *b*, sharing the same vocabulary with stemmed terms excluding stop words. Blocks in the training set are used for PLSA parameter estimation as described in Section 2 with a preset topic number *T*. This model fitting phrase yields two parameters: $P(w|z)$ as term distribution over a certain latent topic z, and $P(z|b)$ as topic distribution over training blocks. The former is used to perform a *folding-in* process to get the representative $P(z|b')$ for other new blocks from the testing set. The folding-in procedure mentioned in Section 2 is performed on each new block b' to compute the topic distribution $P(z|b')$ for all the latent variable z. The estimated distribution of words for each b', $P(w|b')$, is then calculated as:

$$P(w|b') = \sum_z P(w|z)P(z|b') \quad (6)$$

$P(w|b')$ is later used in lexical cohesion measure to compare text blocks.

4. Story segmentation algorithm

4.1. Lexical cohesion measure

In a lexical cohesion based method for story segmentation, a cohesion indicator is required to imply the semantic variation in text. Cosine similarity is a measure of the closeness between two vectors and has been widely used as a similarity metric in information retrieval and text classification/segmentation.

Given the vectorized representation of two text blocks b_i and b_j, the cosine similarity between them can be calculated as:

$$Sim(i,j) = \cos(b_i, b_j) = \frac{\sum_t v_{i,t} v_{j,t}}{\sqrt{\sum_t v_{i,t}^2 \sum_t v_{j,t}^2}} \quad (7)$$

where *t* ranges over all terms in the vocabulary, and $v_{i,t}$ is a weighted value assigned to term *t* in block b_i.

After transforming the word-document co-occurrence observations to the latent semantic layer by PLSA, distributions of words over documents are obtained. Cross entropy is a divergence measure based on the Shannon Entropy and has been employed to depict how different two probabilistic distributions are and thus can represent the lexical score between blocks. The cross entropy for two discrete distributions *p* and *q* over random variable *X* with possible value x is defined as:

$$H(p,q) = -\sum_x p(x) \log q(x) \quad (8)$$

This measure gets its minimum when $p = q$. We apply Eq.(8) to define the difference measure between block b_i and b_j:

$$CrossEnt(i,j) = -\sum_w P(w|b_i) \log P(w|b_j) \quad (9)$$

where $P(w|b_i)$ is the distribution of all words for b_i calculated using Eq.(6). Finally, we normalize the dissimilarity measure to a non-negative value less than 1 as:

$$Dissim(i,j) = \frac{CrossEnt(i,j) - CrossEnt(i,i)}{CrossEnt(i,j)} \quad (10)$$

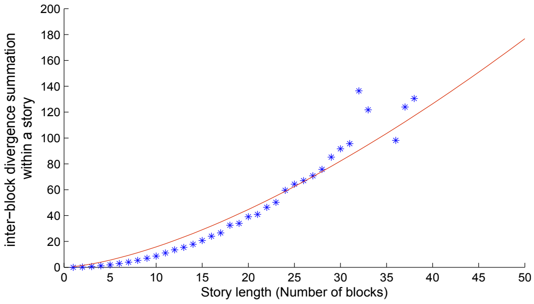

Figure 1: Empirical distribution of intra-story divergence against story length in TDT2 VOA English corpus and fitted l^{α} curve (red line).

4.2. Story boundary identification

An intuitive scheme for story boundary identification is to locate valleys or peaks on the sequence of lexical scores between adjacent blocks. Such implementations include the typical TextTiling method. In the TextTiling method, lexical score is calculated between each consecutive block pair b_l and b_{l+1} according to Eq.(7) and story boundaries are identified at the inter-block positions where the lexical scores are lower than a preset threshold θ. When cross entropy is applied in TextTiling, we calculate the lexical score of two adjacent blocks b_l and b_{l+1} using Eq.(10) and those inter-block positions with a dissimilarity over a preset threshold θ are considered story boundaries.

The TextTiling-like method takes account of the local divergence of a text, which performs better when there are salient story topic changes in lexical distribution. However, sometimes topic transitions between two adjacent news are smooth and distributional variations are subtle. Therefore we use a dynamic programming (DP) algorithm to obtain the global optimal solution, which can effectively catch the smooth topic shifts.

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_K\}$ denote a hypothesis segmentation of document $\mathcal{D} = \{b_1, b_2, \ldots, b_n\}$, which divides $\mathcal{D}$ into K stories. The optimization target of the proposed DP method is to minimize the cost of a specific $\mathcal{S}$. The cost of grouping a number of blocks into one segment s_k is represented by the total dissimilarities between blocks in s_k:

$$cost_{u\leftrightarrow v} = cost(s_k) = \frac{\sum_{i=u}^{v-1}\sum_{j=i+1}^{v} D(i,j)}{N(len(s_k))} \tag{11}$$

where u and v are the first and last block of s_k, and $D(i,j)$ is defined as $Dissim(i,j)$ in Eq.(10) for cross entropy measure and $1-Sim(i,j)$ using Eq.(7) for cosine similarity measure, respectively. $N(len(s_k))$ is a normalization factor where $len(s_k)$ is the number of blocks in segment s_k.

Eq.(11) takes both the intra-segment lexical divergence and segment length into account. Generally, the summation of inter-block divergence takes a larger value when there are more blocks in the segment. Hence, shorter segments are preferred and the optimal segmentation tends to achieve many small segments whereas story length varies in real-world broadcast news programs. In order to make the cost of short and long segments comparable, a normalization step is introduced. As shown in Figure 1, the distribution of inter-block divergence within a story over story length approximately follows a power function. We employ this prior information as the normalization factor:

$$N(l) = l^{\alpha}, \alpha > 1 \tag{12}$$

where l is the segment length and α functions as a suppression rate parameter and is empirically tuned.

The total cost of a possible segmentation $\mathcal{S}$ is defined as the sum of all the K segments cost functions:

$$C(S) = \sum_{k=1}^{K} cost(s_k) \tag{13}$$

Finally, we come to the following optimization problem in order to seek the optimal segmentation $\hat{\mathcal{S}}$:

$$\hat{S} = \arg\min_{S} C(S) \tag{14}$$

The minimization of Eq.(13) can be achieved with the following recursive formulas:

$$f(k,i) = \min_{k\le j\le i}\{f(k-1,j-1) + cost_{j\leftrightarrow i}\}, 1 < k \le K \tag{15}$$

$$b(k,i) = \arg\min_{k\le i} f(k,i), 1 \le k \le K \tag{16}$$

$$\text{s.t. } f(1,i) = cost_{1\leftrightarrow i}, 1 \le i \le n \tag{17}$$

where $cost_{j\leftrightarrow i}$ is the cost function defined by Eq.(11), $f(k,i)$ is the minimal cost of segmenting the first i blocks in $\mathcal{D}$ into k segments, and $b(k,i)$ is a back-point table used to recover the optimal segmentation $\hat{\mathcal{S}}$.

5. Experiments

5.1. Experimental setup

To evaluate the proposed approaches, story segmentation is performed on TDT2[1] VOA English broadcast news corpus. We experimented on the LVCSR transcripts of the BN recordings, with manually annotated story boundary information. The 111 news programs of the corpus are separated to three non-overlapping sets: a training set of 56 files for PLSA model estimation, a development set of 27 files for empirical parameters tuning and a test set of 28 files for performance evaluation.

We carried out story segmentation experiments with four PLSA based methods, namely:

- PLSA-DP-CE, which uses dynamic programming for boundary identification and uses cross entropy for lexical cohesion measure;
- PLSA-DP-CS, which uses dynamic programming for boundary identification and uses cosine similarity for lexical cohesion measure;
- PLSA-TT-CE, which uses TextTiling for boundary identification and uses cross entropy for lexical cohesion measure;
- PLSA-TT-CS, which uses TextTiling for boundary identification and uses cosine similarity for lexical cohesion measure.

Classical TextTiling without using any latent semantic analysis techniques and LSA based TextTiling [12] were tested for comparison. When DP was applied, the number of stories (K) was provided as a priori. In some preliminary experiments related to [13] we found that prior K is not sensitive to the segmentation performance in TextTiling-based systems, so prior K is not provided when we apply TextTiling in this paper. The phoneme n-gram sequences were generated from the word transcripts using the CMU dictionary. The evaluation criterion used is *F1-measure*, i.e., the harmonic mean of *recall* and *precision*. According to the TDT2 standard, a detected boundary is considered correct if it lies within a 15-seconds tolerant window on each side of a reference boundary.

[1] http://www.ldc.upenn.edu/Projects/TDT2

Table 1: Experimental results in terms of F1-measure on word and phoneme n-gram levels.

Approach	word	phoneme			
	unigram	unigram	bigram	trigram	quadgram
PLSA-DP-CE	0.6815	0.4238	0.5729	0.6985	0.6718
PLSA-DP-CS	0.6759	0.4530	0.5591	0.6718	0.6345
PLSA-TT-CE	0.6014	0.4976	0.5462	0.6379	0.6154
PLSA-TT-CS	0.5936	0.4665	0.5292	0.6024	0.6207
LSA-TT-CS	0.5439	0.4680	0.5034	0.5206	0.5393
Classical TT	0.5341	0.4752	0.5035	0.5349	0.5258

DP: Dynamic programming, CE: Cross entropy measure, CS: Cosine similarity measure, TT: TextTiling.

For each method under evaluation, empirical tuning was performed on the development set to select optimal parameter settings achieving the highest F1-measure. Then we applied the best-tuned parameters on the test set. Tuning parameters in TextTiling include block length, sliding window shift, lexical score threshold while tuning parameters in DP include suppression rate α in l^{α} and block length.

5.2. Experimental results and analysis

Table 1 summarizes the experimental results in terms of F1-measure on the test set. We can observe that in general the four methods using PLSA (PLSA-DP-CE, PLSA-DP-CS, PLSA-TT-CE, PLSA-TT-CS) notably outperform the other two approaches in each level. For instance, the PLSA-TT-CE approach achieves a relative gain of 19.26% (from 0.5349 to 0.6379) over classical TextTiling on phoneme trigram. This indicates that applying PLSA to story segmentation task can effectively improve the segmentation performance, owing to its reflection of lexical relations beneath the actual word occurrences in text. On the contrary, LSA provides less improvement than PLSA against original TextTiling, which may be explained by the inadequacies of LSA such as its inability to capture the multiple meanings of words. The significant performance gain from LSA-TT-CS to PLSA-TT-CS implies the superiority of PLSA over LSA.

The performance gain from using a different lexical cohesion measure, i.e., the divergence measure based on cross entropy instead of cosine similarity, also draws our attention. For example, phoneme quadgram PLSA-DP-CE method improves 5.88% (from 0.6375 to 0.6718) relatively comparing to the corresponding PLSA-DP-CS. We believe this shows the advantage of cross entropy in describing distributional variation. Since the original word/subword counts are replaced by probabilistic statistics which reveal the underlying semantic meaning of the contents, it is more suitable to adopt measures of comparing distributions. It is also noticed that PLSA-DP-CE significantly outperforms PLSA-TT-CE with a relative improvement of 13.32% (from 0.6014 to 0.6815) when word unigram is used. This can be interpreted by the characteristic of DP that it considers global semantic variations and offers an optimal solution.

Comparing the results using word unigram and phoneme n-grams, phoneme trigram and quadgram offer overall better performances than word unigram. In PLSA-TT-CE and PLSA-DP-CE, phoneme trigram provides relative improvements of 6.07% (from 0.6014 to 0.6379) and 2.49% (from 0.6815 to 0.6985) respectively over word unigram. These results suggest that although words are more distinctive than phonemes, a proper n-gram conveys sufficient semantic information to distinguish from others and offers competitive capability to handle OOV problems. The inferior performances of phoneme unigram and bigram may be due to their smaller number of n-gram entries which lead to lower discriminative capacities.

6. Conclusions

This paper investigates the use of PLSA for broadcast news story segmentation. To address the OOV problem brought by LVCSR, we conduct story segmentation based on phoneme units. A divergence measure which calculates inter-block cross entropy is adopted, and it is compared with cosine similarity for lexical cohesion measure. We further evaluate DP for story boundary identification. Experimental results suggest that i) PLSA can effectively boost story segmentation performance; ii) cross entropy is a promising measurement to depict distribution disparity; iii) the proposed DP solution, which benefits from the overall optimization property, provides the best performance.

7. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (60802085), the China MOE Program for New Century Excellent Talents in University (2008) and the Natural Science Basic Research Plan of Shaanxi Province (2011JM8009).

8. References

[1] M. A. Hearst, “TextTiling: segmenting text into multi-paragraph subtopic passages,” *Comput. Linguist.*, vol. 23, no. 1, pp. 33–64, 1997.

[2] P. Fragkou, V. Petridis, and A. Kehagias, “A dynamic programming algorithm for linear text segmentation,” *J. Intell. Inf. Syst.*, vol. 23, pp. 179–197, 2004.

[3] I. Malioutov and R. Barzilay, “Minimum cut model for spoken lecture segmentation,” in *Proc. of ACL*, 2006, pp. 25–32.

[4] F. Choi, P. W.-Hastings, and J. Moore, “Latent semantic analysis for text segmentation,” in *Proc. of EMNLP*, 2001, pp. 109–117.

[5] T. Hofmann, “Unsupervised learning by probabilistic latent semantic analysis,” *Mach. Learn.*, vol. 42, pp. 177–196, 2001.

[6] T. Brants, F. Chen, and I. Tsochantaridis, “Topic-based document segmentation with probabilistic latent semantic analysis,” in *Proc. of CIKM*, 2002, pp. 211–218.

[7] X. Wang, L. Xie, B. Ma, E.-S. Chng, and H. Li, “Phoneme Lattice based TextTiling towards Multilingual Story Segmentation,” in *Proc. of Interspeech*, Makuhari, Japan, 2010.

[8] J. E. Shore, “Minimum Cross-Entropy Spectral Analysis,” *IEEE Trans. on ASLP*, vol. 29, no. 2, pp. 230–237, 1981.

[9] W. J. Teahan, “Text classification and segmentation using minimum cross-entropy,” in *Proc. of RIAO*, 2000.

[10] R. Nallapati, “The smoothed dirichlet distribution: Understanding cross-entropy ranking in information retrieval,” Ph.D. dissertation, Univ. of Massachusetts Amherst, 2006.

[11] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proc. of SIGIR*, 1999, pp. 50–57.

[12] Y. Yang and L. Xie, “Subword Latent Semantic Analysis for Texttiling-Based Automatic Story Segmentation of Chinese Broadcast News,” in *Proc. of ISCSLP*, Kunming, China, 2008.

[13] L. Xie, L. Zheng, Z. Liu, and Y. Zhang, “Laplacian eigenmaps for automatic story segmentation of broadcast news,” *IEEE Trans. Audio, Speech, and Language Processing*, 2011.