J Ambient Intell Human Comput (2017) 8:925–936
DOI 10.1007/s12652-017-0501-9

ORIGINAL RESEARCH

A hybrid neural network hidden Markov model approach for automatic story segmentation

Jia Yu[1,2] · Lei Xie[1] · Xiong Xiao[3] · Eng Siong Chng[3]

Received: 6 February 2017 / Accepted: 29 April 2017 / Published online: 11 May 2017

Abstract We propose a hybrid neural network hidden Markov model (NN-HMM) approach for automatic story segmentation. A story is treated as an instance of an underlying topic (a hidden state) and words are generated from the distribution of the topic. The transition from one topic to another indicates a story boundary. Different from the traditional HMM approach, in which the emission probability of each state is calculated from a topic-dependent language model, we use deep neural network (DNN) to directly map the word distribution into topic posterior probabilities. DNN is known to be able to learn meaningful continuous features for words and hence has better discriminative and generalization capability than n-gram models. Specifically, we investigate three neural network structures: a feed-forward neural network, a recurrent neural network with long short-term memory cells (LSTM-RNN) and a modified LSTM-RNN with multi-task learning ability. Experimental results on the TDT2 corpus show that the proposed NN-HMM approach outperforms the traditional HMM approach significantly and achieves state-of-the-art performance in story segmentation.

Keywords Neural network · Long short-term memory · Hidden Markov model · Multi-task learning · Story segmentation · Topic modeling

1 Introduction

The development of multimedia and web technologies has triggered exponential growth of multimedia collections, such as broadcast news, lectures, and meeting recordings. With the vast amount of multimedia contents, there are increasing demands for multimedia processing technologies, such as story segmentation, topic detection and tracking (James 2002; Fiscus et al. 1999), document summarization (Rau et al. 1989), content indexing and retrieval (Lee and Chen 2005) and information extraction (Soderland 1999). Serving as an important precursor, the task of story segmentation (James 2002; Beeferman et al. 1999; Reynar 1994; Hearst 1997) aims to partition a stream of video, audio or text into a sequence of topically coherent segments, each of which addressing a central topic.

Story segmentation has been historically studied for diverse genres, such as broadcast news programs (Rosenberg and Hirschberg 2006; Chen et al. 2016), meeting recordings (Banerjee and Rudnicky 2006) and lectures (Malioutov and Barzilay 2006; Malioutov et al. 2007), etc., over different types of media, including audio (Malioutov et al. 2007; Shriberg et al. 2000; Charlet et al. 2015), video (Chaisorn et al. 2003) and text (Yamron et al. 1998; Beeferman et al. 1999; Sherman and Liu 2008; Van Mulbregt et al. 1998; Hearst 1997; Banerjee and Rudnicky

✉ Jia Yu
jiayu@nwpu-aslp.org

Lei Xie
lxie@nwpu-aslp.org

Xiong Xiao
xiaoxiong@ntu.edu.sg

Eng Siong Chng
aseschng@ntu.edu.sg

[1] Shaanxi Provincial Key Laboratory of Speech and Image Information Processing, School of Computer Science, Northwestern Polytechnical University, Xi'an, China

[2] School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang, China

[3] Temasek Laboratories@NTU, Nanyang Technological University, Singapore, Singapore

2006). In this paper, we aim to perform story segmentation for broadcast news transcripts. Note that, with the recent tremendous success of large vocabulary continuous speech recognition (LVCSR) using deep neural network (DNN) (Yu and Deng 2015; Graves et al. 2013; Abdel-Hamid et al. 2013; Yu et al. 2013; Schultz and Waibel 2001; Damavandi et al. 2016; Bourlard and Morgan 2012), we can easily obtain high accuracy transcripts for broadcast news and traditional text segmentation approaches can be easily applied to speech recognition transcripts of broadcast news.

Story segmentation on text usually depends on the phenomenon of lexical cohesion, i.e., words in the same story are topically or semantically similar while words in different stories are quite different in topic or semantic distributions. Thus, word and sentence representations, which catch semantic or topic information, are essential for story segmentation. Bag-of-words (BOW) representation, or typically term frequency–inverse document frequency (*tf–idf*), is a simple-but-effective representation used in story segmentation approaches, e.g., TextTiling and dynamic programming (DP) (Hearst 1997; Xie et al. 2011; Bouchekif et al. 2014). However, *tf–idf* only counts the appearances of words in each sentence and does not take inter-word semantic relations into account. Instead, probabilistic topic models, e.g., probabilistic latent semantic analysis (pLSA) (Lu et al. 2011a), latent Dirichlet allocation (LDA) (Blei et al. 2003), and LapPLSA (Lu et al. 2011b), employ latent variables, namely topics, to reveal the salient statistic patterns in the co-occurrence of words, catching intrinsic semantic relations between words. With these probabilistic models, BOW based word representations are transformed into topic representations and significant improvements have been achieved in story segmentation performance (Blei et al. 2003; Hofmann and Thomas 1999). Alternatively, artificial neural networks (ANN) can be used to model topics and some neural topic models have shown promising performances in tasks like document classification (Wan et al. 2012; Li et al. 2016; Lai et al. 2015), document retrieval (Larochelle and Lauly 2012) and topic detection (Kumar and FD'Haro 2015).

Using the above word/sentence representations, story segmentation approaches can be categorized into detection-based and probabilistic-model-based approaches. The former approaches find optimal partitions over the word sequence by optimizing a local objective, e.g., in TextTiling (Hearst 1997; Xie et al. 2011), or a global objective, e.g., in NCuts (Lu et al. 2011b) and DP approaches (Fragkou et al. 2004; Heinonen 1998; Xie et al. 2012). The latter statistical model based approaches assign words or sentences with latent topic variables and the shift of the latent variable assignments indicate a story boundary. Popular such approaches include PLSA (Hofmann and Thomas 1999), BayesSeg (Eisenstein and Barzilay 2008) and dd-CRP (Yang et al. 2014).

As a powerful probabilistic sequential labeling tool, hidden Markov model (HMM) (Rabiner and Juang 1986) has been successfully introduced to the story segmentation task (Sherman and Liu 2008; Van Mulbregt et al. 1998; Yamron et al. 1998). In these approaches, each HMM hidden state is regarded as an underlying topic and words are generated from the distribution of topics. Naturally, the switch from one hidden state to another indicates a story boundary. Transition and emission probabilities can be inferred from a training corpus. Specifically, the emission probability of a state is calculated from a topic-dependent language model (LM), while the transition probability is determined by a development set. The Viterbi algorithm is used to decode the story boundaries from an input text sequence and the position of topic change is thus regarded as a story boundary.

In this paper, we propose a hybrid neural network hidden Markov model (NN-HMM) approach for story segmentation. Unlike the topic-dependent LM used in traditional HMM-based approaches (Van Mulbregt et al. 1998; Yamron et al. 1998), which is a generative model of the word sequence, we use a neural network to directly map the word observation into topic posterior probabilities. Deep neural networks (DNN) are known to be able to learn meaningful continuous features for words. Hence we believe that they have better discriminative and generalization capabilities than n-gram models (Haidar and kurimo 2016; Chunwijitra et al. 2016). As the neural network architecture can be quite flexible, we have studied three different structures for topic posterior prediction.

- DNN: A feed-forward neural network takes as input the BOW vector computed by a context with a fixed window size of words;
- LSTM: An recurrent neural network (RNN) with long short-term memory (LSTM) cells models the contextual information in the model structure;
- LSTM-MTL: The LSTM-RNN is modified with multi-task learning (MTL) ability, i.e., besides topic posterior output, auxiliary output is added to predict word identity as a language model.

Since the feed-forward network has limited ability in modeling context, a long feature window in the input is usually used to cover some contextual information. In contrast, a recurrent network directly uses an acyclic connection to naturally model the contextual information in the model structure itself. Hence we believe that LSTM-RNN is more promising in topic modeling in which contextual information is essential. Previous studies show that through related auxiliary tasks, MTL can improve the generalization ability

of the main task (Yu and Deng 2015). In this study, we use an N-gram language model as the auxiliary task that predicts the next word given a sequence of previous words. Neural network language model (NNLM) can learn important semantic information (Mikolov et al. 2013b; Le and Mikolov 2014) and this task is highly related to topic label prediction. The proposed approaches are evaluated on the TDT2 corpus and results show that the proposed approach is able to achieve state-of-the-art performance in story segmentation.

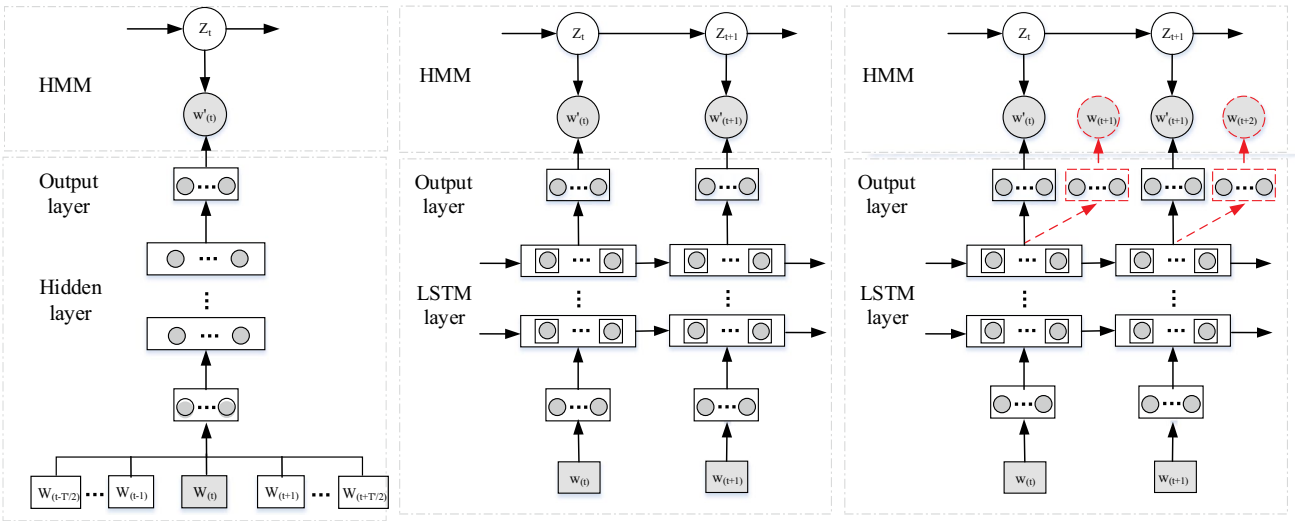
2 The proposed approach

Figure 1 shows the architectures of the proposed NN-HMM approach for story segmentation. The upper part of the architectures is an HMM, in which each state represents an underlying topic and associates with an emission probability. The transition probabilities among states, used to model the switching between stories, are represented by a transition matrix. The neural networks, as shown in the lower part of the architectures, construct a topic model and generate topic posterior. The topic posterior is further converted to likelihood and used for Viterbi decoding that determines the story changes.

As the neural network architecture can be quite flexible, we have investigated three different structures for topic modeling. In Fig. 1a, namely DNN-HMM, an ordinary feed-forward neural network takes as input the BOW vector computed by a context with a fixed window size of words ($T+1$). Through non-linear transformations, the neural network maps BOW vectors to topic posteriors. Using the topic posterior, the HMM finally finds the topic labels of the input word sequence. The position of changing topic label is regarded as a story boundary.

In DNN-HMM, the BOW vector fed to the neural network is obtained by summing the BOW vector representations of the words in the sliding window. As the feed-forward network has limited ability in modeling context, a long feature window in the input is usually used to cover some contextual information (Yu and Deng 2015). But this feed-forward structure can just model a limited number of context words and also lose the important word order information which is known to be important for modeling the topic distribution (Tian et al. 2016; Ghosh et al. 2016). Recurrent neural network (RNN) instead uses an acyclic connection to naturally model the contextual/sequential information in the model structure itself. Hence, in this study, RNN with long short-term memory (LSTM) cells is investigated in topic modeling for story segmentation, as shown in Fig. 1b. As we know, LSTM-RNN has been successfully used in many sequential modeling tasks including language modeling (Sundermeyer et al. 2012), acoustic modeling (Tan et al. 2016) and punctuation prediction (Xu et al. 2016).

As a neural network language model (NNLM) can capture semantic information (Mikolov et al. 2013a; Le and Mikolov 2014) that may benefit topic posterior prediction, we use multi-tasking learning (Collobert and Weston 2008; Seltzer and Droppo 2013)to incorporate an NNLM as the



(a) DNN-HMM **(b)** LSTM-HMM **(c)** LSTM-MTL-HMM

Fig. 1 The architectures of the proposed NN-HMM approach for story segmentation. The *upper part* of the architectures is an HMM and the *lower part* is either a feed-forward neural network (DNN-HMM in **a**) or a recurrent neural network with long short-term memory cells (LSTM-HMM in **b**), or an LSTM-RNN with an n-gram language model as the second task (MTL-HMM)

second task in the neural network based topic model. As shown in Fig. 1c, we add an additional output (in red) to the last LSTM layer to predict the probability of next word given a sequence of previous words. The additional output is actually an N-gram LM which helps to infer the parameters in the training process and can be discarded in testing. The original output of the neural network is the same as the the LSTM model in Fig. 1b, which is the topic posterior used for HMM decoding.

3 HMM for story segmentation

HMM is a typical generative model historically used for story segmentation (Van Mulbregt et al. 1998; Yamron et al. 1998), in which each hidden state represents a topic. The words, called observations, are generated from these topics following certain distribution. The transition between hidden states is modeled by an $N \times N$ matrix which can be learned from a set of training data. Each hidden state is associated with a probability distribution function (PDF) that models the N-gram word distribution for the topic represented by the state. For example, in Sherman and Liu (2008), a topic-dependent unigram language model is used as the emission probability of each HMM state.

Given a sequence of observation words and the trained HMM, the topics are inferred through an optimization process:

$$\hat{z} = \arg\max_{z} p(z|\boldsymbol{w};\theta), \tag{1}$$

where $z = [z_1, z_2, \ldots, z_T]$ is the inferred topic sequence and $\mathbf{w} = [w_1, w_2, \ldots, w_T]$ is the sequence of T observed words. θ represents HMM parameters including transition probability and state emission PDFs. By the Bayesian rule, the above optimization problem is equivalent to:

$$\hat{z} = \arg\max_{z} p(\boldsymbol{w}|z;\theta)p(z)/p(\boldsymbol{w}), \tag{2}$$

$$= \arg\max_{z} p(\boldsymbol{w}|z;\theta)p(z). \tag{3}$$

In the optimization process, $p(\boldsymbol{w})$ can be ignored as it does not depend on z. $p(z)$ is the transition probability between states and it can be calculated as follows:

$$p(z) = p(z_1)\prod_{t=2}^{T} p(z_t|z_{t-1}), \tag{4}$$

where $p(z_t|z_{t-1})$ is the transition probability from state z_{t-1} to z_t. We have the assumption that the words in neighboring time steps are independent given the state sequence, and hence

$$p(\boldsymbol{w}|z) = \prod_{t=1}^{T} p(w_t|z_t), \tag{5}$$

where $p(w_t|z_t)$ is the conditional distribution of words given the topic, which is a topic-dependant LM. Equation (5) only allows unigram topic LM to be used. To use higher order for N-gram LM, we can use a fixed window of words (Blei and Moreno 2001) or sentence (Blei and Moreno 2001) as the basic observation unit.

The transition probability and topic-dependant LM can be calculated from a training corpus which is composed of segmented stories with boundaries and annotated topic labels. If the topic label is not available, we can cluster the stories into predefined number of topics. Followed the Eqs. from (3) to (5), we can use the Viterbi algorithm to find the optimal topic sequence for test data efficiently.

4 Neural networks for topic posterior prediction

Neural networks have strong capacity to transform the raw feature into more meaningful feature representation that is more suitable for the target task (Mikolov et al. 2013a, b). In this paper, we use neural networks to construct topic models and generate topic posteriors from original BOW vectors. Then the posterior is converted to likelihood and used in Eq. (5) for Viterbi decoding in the HMM framework. Note that the NN generated topic posteriors can be directly used in other story segmentation approaches, e.g. TextTiling (Hearst 1997; Xie et al. 2011) and DP (Fragkou et al. 2004; Heinonen 1998; Xie et al. 2012).

4.1 DNN for topic posterior prediction

An ordinary deep neural network is actually a multi-layer perception (MLP), i.e., a feed-forward neural network model that maps sets of input data onto a set of outputs. In our case, the input and output are BOW features and topic posteriors, respectively. An MLP usually consists of an input layer, a hidden layer and an output layer, and the nodes in each layer are fully connected to the nodes in another layer. A DNN generalizes an MLP with multiple hidden layers. A DNN also can be considered as a hierarchical feature learner with nonlinear transformations refining the input representation to a better one, which is topic posterior in our case.

As different topics employ different word distributions and the topic information can be embedded in the context, the BOW representation of current word w_t with local context is computed as

$$\boldsymbol{x}_t = \frac{1}{T'+1}\sum_{\tau=-T'/2}^{T'/2} \tilde{w}_{t-\tau}, \tag{6}$$

where T' is the window length of the context, and w_t represents the current word encoded by one-hot representation. $\boldsymbol{x}_t$ is the BOW vector of current word, which is computed by averaging its context words and has the dimension same with the size of the vocabulary. In the beginning and ending of the sentence, we replace T' with the actual number of words in the window, and thus $\boldsymbol{x}_t$ is properly normalized despite of its position in the sentence.

The BOW feature $\boldsymbol{x}_t$ is nonlinearly transformed by the hidden layers of the DNN and the output of lth hidden layer is computed by

$$\boldsymbol{h}_l = f_l(\boldsymbol{W}_l \boldsymbol{h}_{l-1} + \boldsymbol{b}_l), \tag{7}$$

where $\boldsymbol{h}_l, f_l, \boldsymbol{W}_l$ and $\boldsymbol{b}_l$ are the output, activation function, transform matrix and bias vector at layer l, respectively. In this model, we use sigmoid as activation function and the hidden layers are fully connected. The input of first hidden layer $\boldsymbol{h}_0 = \boldsymbol{x}_t$. Given the input, the posterior probability of the ith topic is

$$p(z_t = i|\boldsymbol{x}_t) = \frac{e^{h_L(i)}}{\sum_{j=1}^{J} e^{h_L(j)}}, \tag{8}$$

where $h_L(i)$ is the ith element of the last hidden layer's output. J is the total number of topic classes. We use softmax function to generate the topic posterior probability.

Training the DNN topic model (TM) aims to optimize the objective function:

$$\mathcal{L}_{\mathrm{TM}} = \sum_{t}^{T} \log p(z_t|\boldsymbol{x}_t), \tag{9}$$

where T is the total number of training samples and $\boldsymbol{x}_t$ is the input feature. Training is achieved by error back propagation (BP) (Li et al. 2012) on a training set. Specially, the training process includes forward propagation and backward propagation. The forward propagation calculates the prediction errors (i.e. the topic posterior prediction error) and the backward propagation reversely passes the errors back to modify the model parameters.

4.2 LSTM for topic posterior prediction

Context information is a critical factor for topic modeling (Ghosh et al. 2016). Allowing cyclical connections in a feed-forward neural network, we obtain recurrent neural networks (RNNs) (Williams and Zipser 1989; Haidar and kurimo 2016). Different from the feed-forward networks that consider contextual information by windowing the input, RNNs are able to directly incorporate contextual information from previous input vectors, which allows them to remember past inputs and persist in the network's internal state. This property makes them an attractive choice for topic posterior prediction.

LSTM-RNN is a special recurrent neural network that is composed of units called memory blocks in the hidden layer, as shown in Fig. 2. The memory blocks contain memory cells with self-connected pipeline storing the temporal state of the cell. The multiplicative units, called gates, modify the state of memory cell and control the input and output of the memory cell given the input and output of previous time steps. The forget gates determine what kind of information should be discarded from the state of memory cell by scaling the internal state of the cell before adding it as input to the cell through the self-recurrent connection. The input gates control the information flow into the memory cell and the output gates control the information flow to the rest of neural network. In modern LSTM architecture, there are peephole connections from its internal cells to the gates in the same cell to learn precise value of outputs.

An LSTM-RNN network maps an input sequence $\boldsymbol{x} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T]$ to a output topic posterior sequence $z = [z_1, \ldots, z_T]$ using the following equations iteratively from time step $t = 1$ to T:

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{ic}\boldsymbol{c}_{t-1} + \boldsymbol{b}_i), \tag{10}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{fx}\boldsymbol{x}_t + \boldsymbol{W}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{fc}\boldsymbol{c}_{t-1} + \boldsymbol{b}_f), \tag{11}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot g(\boldsymbol{W}_{cx}\boldsymbol{x}_t + \boldsymbol{W}_{ch}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c), \tag{12}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{ox}\boldsymbol{x}_t + \boldsymbol{W}_{oh}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{oc}\boldsymbol{c}_t + \boldsymbol{b}_o), \tag{13}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \boldsymbol{p}(\boldsymbol{c}_t), \tag{14}$$

where $\boldsymbol{x}_t$ is a vector converted from BOW representation of word w_t through a projection layer. $\boldsymbol{i}$, $\boldsymbol{o}$, $\boldsymbol{f}$ and $\boldsymbol{c}$ denote input gate, output gate, forget gate and memory cell vectors,

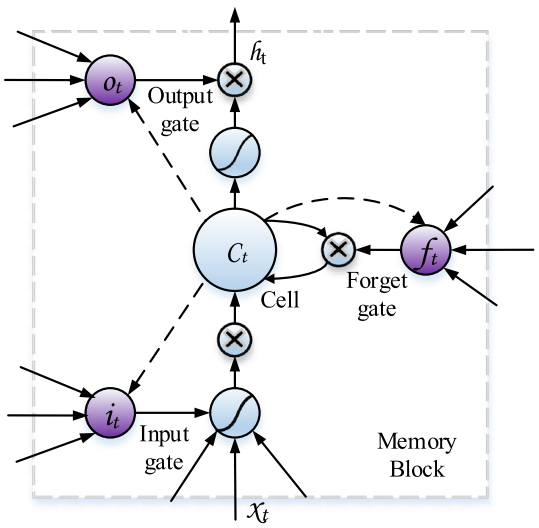


Fig. 2 The architecture of the memory block used in LSTM-RNN

respectively. $\boldsymbol{W}$ is the weight matrix, e.g., $\boldsymbol{W}_{xi}$ is the weight matrix between input gate and input. $\boldsymbol{W}_{ic}, \boldsymbol{W}_{fc}, \boldsymbol{W}_{oc}$, represented by diagonal weight matrix, are peephole connection. The $\boldsymbol{b}$ terms denote the bias vector, e.g., $\boldsymbol{b}_i$ is the input gate vector. σ is the logistic sigmoid activation function. $\boldsymbol{g}$ and $\boldsymbol{p}$ are cell input and cell output activation functions, which is tan$h$ in this paper. $\odot$ is the element-wise product operator. $\boldsymbol{h}_t$ is the output of LSTM layer and used as input in Eq. (8) to compute topic posteriors.

Training the LSTM-RNN is achieved by the error back propagation through time (BPTT) algorithm (Werbos 1988; Cullar et al. 2005), which propagates errors through long time steps and BP is iteratively used to modify model parameters in proportion to its derivative with respect to the errors.

4.3 Multi-task learning for topic posterior prediction

Multi-task learning (MTL) is a commonly-used machine learning strategy in which a primary learning task is resolved jointly with *related* task(s) using a shared input stream (Seltzer and Droppo 2013; Wu et al. 2015; Liu et al. 2015; Huang et al. 2015). A well chosen secondary task can help improve the performance of the main task in the training process, while in the testing process, the secondary task can be safely discarded.

According to Yu and Deng (2015), the key to the success of MTL is that the main and auxiliary tasks need to be related. Here related does not mean the tasks have to be similar. Instead, it means at some level of abstraction these tasks share part of the representation. Learning them together can help constrain the possible functional space of each task and thus improve the generalization ability of each task. Neural networks are well suited to support MTL (Yu and Deng 2015) as different tasks can easily share the hidden layers that learn hierarchical feature representations. In this paper, under a shared neural network architecture, the primary task is to construct a topic model to predict the current word's topic label, while the secondary task is an N-gram language model that predicts the next word given a sequence of previous words, as shown in Fig. 1c. Plenty of evidence has shown that neural network language model (NNLM) can learn important syntactic and semantic information from raw texts (Mikolov et al. 2013a; Le and Mikolov 2014). The distributed representation, i.e., the so-called word vector or word embedding, has been successfully used in many natural language processing tasks (Kim 2014; Chen and Manning 2014). Apparently, semantic information is essential in topic label prediction. Thus we believe that word prediction (i.e., N-gram language model) and topic prediction are highly related tasks and they can be integrated in a single neural network under the MTL scheme.

Based on the model structure in Fig. 1b, we simply add an additional output layer following the LSTM layers to predict the next word (the second task). A softmax activation function is used to compute the likelihood for each word in the vocabulary. Specifically, the aim of the secondary task is to optimize the following LM objective function:

$$\pounds_{\mathrm{LM}} = \sum_{t}^{T} \log p(w_t | w_{t-1}, w_{t-2}, \ldots, w_{t-N+1}) \tag{15}$$

where T is the total number of training samples and N is the number of words in the context.

To train the MTL neural network, we maximize the following integrated objective function:

$$\pounds = \pounds_{\mathrm{TM}} + \alpha \pounds_{\mathrm{LM}}, \tag{16}$$

where $\pounds_{TM}$ and $\pounds_{LM}$ are defined in Eqs. (9) and (15), and α is a scalar between 0 and 1. The optimal α is determined by tuning on a development set. In the testing process, the secondary task is simply ignored and we only use the primary task to generate topic posteriors.

4.4 Visualization of the topic posteriors

Figure 3 shows the quality of the predicted topic posteriors of one news program in the training set of the TDT2 corpus (Fiscus et al. 1999). Horizontal axis denotes the index of words while vertical axis is the topic class. Darker color means high probability. We can clearly see that the predicted topic posteriors in (a), (b) and (c) follow the true topic label in (d) reasonably well. This shows that it is suitable to use neural network to predict word topics. We also can see that (c) has the most similar pattern with the ground truth in (d), showing more promising segmentation performance.

4.5 Converting topic posterior to likelihood

The output of the neural network, Eq. (8) is the topic posterior given the input word, but what we need for Viterbi decoding in an HMM is the likelihood $p(w_t|z_t = i)$ as shown in Eq. (5). We first assume that $p(z_t = i|w_t) = p(z_t = i|\mathbf{x}_t)$, i.e., the topic posterior given a word is the same as the topic posterior given the word's local context. Then the likelihood can be obtained from the Bayesian rule

$$p(w_t|z_t = i) = \frac{p(z_t = i|\boldsymbol{x}_t)p(w_t)}{p(z_t = i)}, \tag{17}$$

where $p(w_t)$ does not depend on the topic class and thus can be ignored in the decoding. $p(z_t = i)$ is the prior probability of the topic class i. Note that the way of converting class posterior to observation likelihood in Eq. (17) has been used widely in hybrid DNN-HMM approaches for

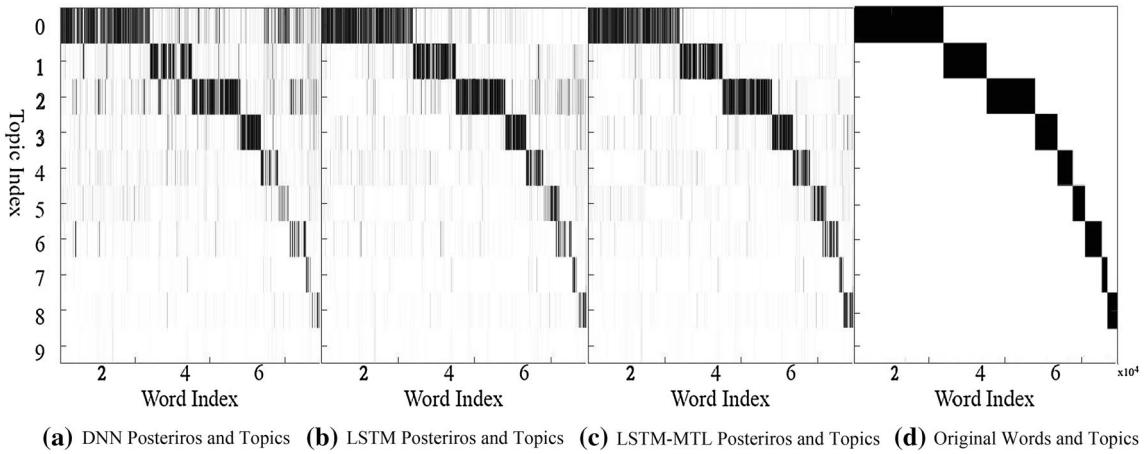


Fig. 3 Predicted topic posterior probabilities versus true topic label for a news program in the training set of the TDT corpus. *Darker* means higher probability. **a**–**c** Plot the topic posteriors of words predicted by DNN, LSTM, and LSTM-MTL models, while **d** is the ground truth with true topic class label of the words

speech recognition (Yu and Deng 2015; Bourlard and Morgan 2012).

4.6 Generating topic labels using clustering

The topic label of each word is usually not readily available. Manual topic labeling is not practical in real applications. In order to get topic labels of words for neural network training, we cluster the segmented stories in the training set to predefined number of clusters using the CLUTO tool (Karypis 2002). The clustering objective is to minimize the inter-cluster similarity and maximize the intra-cluster similarity. After clustering, each story is assigned with a topic label and words in the story has the same topic label. The topic-label words are thus used for neural network training.

The probabilities of word appearance are usually different in different clusters (topics). For example, there is high probability of appearance of words like football, basketball, tennis in a sports news, while bank, stock market and bond appears frequently in economic news. Figure 4 shows the distribution of most frequently appeared words in some selected clusters. From this figure, we can observe that the most frequent words used usually depends on the topic. Such information can be captured by the BOW feature vector and used to predict the topic by a neural network.

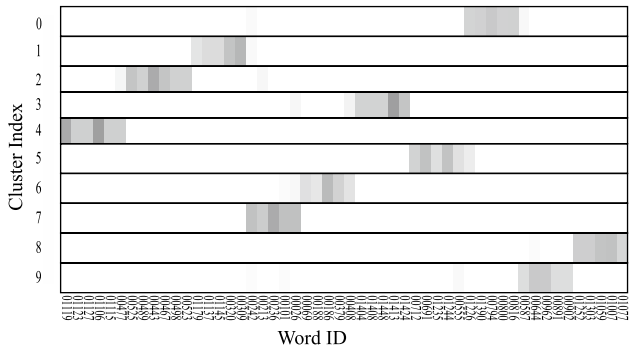


Fig. 4 The distribution of most frequent words in ten clusters. *X-axis* is the index of frequent words in the ten clusters, while *y-axis* is the index of clusters. *Darker color* means higher probability of occurrence

5 Experiments

5.1 Experimental setup

We carried out experiments on the topic detection and tracking (TDT2) corpus (Fiscus et al. 1999) which includes 2280 English broadcast news programs. There are 11, 406 stories in total and each story has an average of 20 topics and 200 words. We construct a vocabulary including 57, 817 words. The corpus was separated into a training set with 1800 programs, a development set and a testing set each with 240 programs. All texts were stemmed by a Porter stemmer and stop words are removed. The CLUTO tool (Karypis 2002) was used to perform clustering on the training set and the topic labels were generated according to the clusters.

For the three kinds of neural networks, we initialized the learning rate to 0.03, and decreased it by a decay rate of 0.999. The value of the momentum was set to 0.9. We used an L2 regularizer and set the value to 3×10^{-6}. The GPU was used to accelerate the training process.

We used F1-measure, i.e., the harmonic mean of recall and precision, to evaluate the story segmentation

performance with a tolerance window of 50 words according to the TDT2 standard (Fiscus et al. 1999). The discovered boundaries were compared to the manually segmented boundaries. Precision is defined as the percentage of declared boundaries that coincide with the referenced boundaries. Recall is defined as the percentage of referenced boundaries that are retrieved. F1-measure is defined as

$$\text{F1-measure} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \quad (18)$$

5.2 Analysis of NN-derived topic posterior features

As we discussed earlier, the neural network derived topic posteriors can be directly used as features. Before testing the proposed hybrid NN-HMM approach, we first examined the performance of the NN-derived topic posterior features. Two typical story segmentation approaches, TextTiling (Hearst 1997) and dynamic programing (DP) (Fragkou et al. 2004), were considered. The NN-derived topic posterior features were compared with *tf–idf* and LDA (Blei et al. 2003) features. Cosine distance is used to compute the similarity score between sentences in the TextTiling approach. Table 1 shows the results of TextTiling and DP on the testing set with different features. The context size of DNN was set to 60 and the number of clusters for LSTM and LSTM-MTL were both set to 150 according to a process of parameters tuning. We can clearly see that NN-derived topic posterior features show superior performances as compared with *tf–idf* and LDA. LSTM outperforms feed-forward DNN, while the best performance is achieved with the help of multi-task learning.

Figure 5 illustrates the sentence similarity matrix dotplots for an episode of broadcast news program from the testing set, in which the similarity is calculated based on *tf–idf*, LDA and the NN-derived topic posteriors, respectively. The red line indicates the real story boundaries. We can see that all dotplot figures contain dark square regions along the diagonal delimited by story boundaries. These regions indicate cohesive topic segments with high sentence similarities. At the meantime, we can see more salient blocks on the topic posterior based dotplots (d, e, f) generated by neural networks and the block patterns are much closer to the ground truth in (a). We also notice that the block pattern provided by LSTM-MTL is the closest to the ground truth with fewer noises.

Table 1 F1-measure of TextTiling and DP approaches on different features

Feature	TextTiling	DP
tf–idf	0.553	0.421
LDA	0.574	0.682
Topic posteriors by DNN	0.663	0.726
Topic posteriors by LSTM	0.682	0.732
Topic posteriors by LSTM-MTL	**0.689**	**0.735**

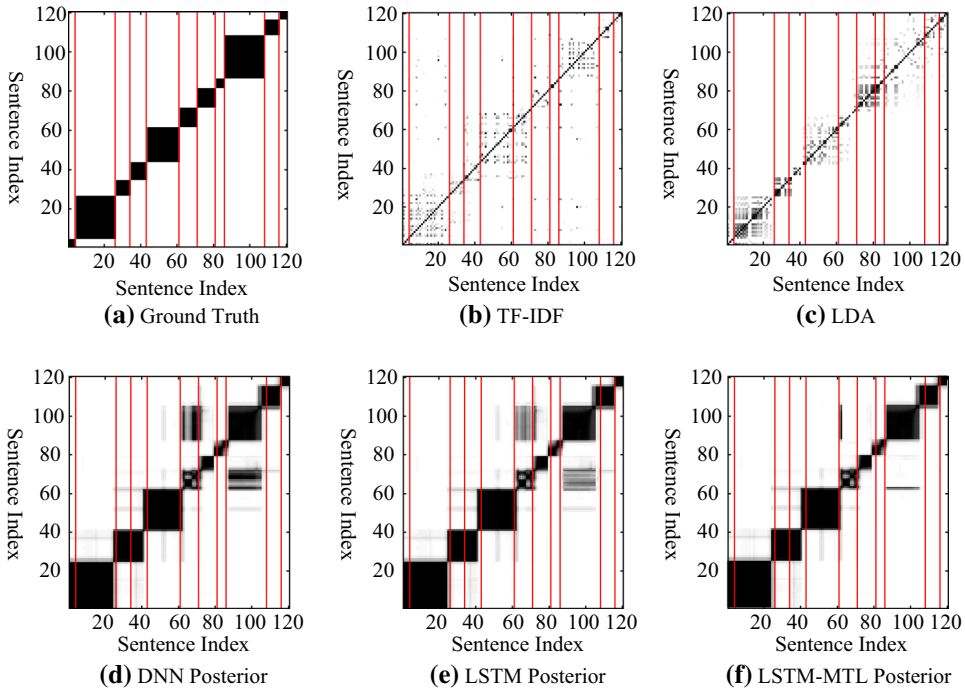


Fig. 5 Sentence similarity matrix dotplots for an episode of broadcast news program from the TDT2 testing set, in which the similarities are calculated based on **b** TF–IDF, **c** LDA and **d** DNN **e** LSTM **f** LSTM-MTL posteriors, respectively. **a** The ideally dotplot drawn with true story boundaries and used as ground truth. *x-axis* and *y-axis* are index of sentences. High similarity values are represented by *dark pixels*. The *vertical lines* indicate the real story boundaries

5.3 Results of DNN-HMM

We trained a DNN with two hidden layers, each of which contains 256 nodes. The context size of the sliding window and the number of topic clusters were empirically tuned on the development set. A diagonal transform and bias vector were used to make the BOW feature vectors to be zero mean and unit variance for the training corpus (global mean and variance normalization). The same transform and bias were also used to normalize BOW vectors in the testing set. We used sentence boundaries to construct sentence unit for the HMM decoding. According to (Sherman and Liu 2008), the self-transition of state is 0.8 (tuned on the development set), while the remaining 0.2 probability is evenly assigned to the switching from the current state to other states.

We first investigated the relationship between F1-measure and the number of clusters (topics), as shown in Table 2. We observe that the F1-measure scores are all above 0.7 for all numbers of clusters tested, from 50 to 200, which suggests the proposed approach is quite stable on varies cluster numbers. We got the highest F1-measure of 0.765 when the cluster number is 170. We also investigated the effect of context size in the DNN-HMM approach. Table 3 shows the results. We got the highest F1-measure when the context size is set to 60 words. The results show that the F1-measure is also not very sensitive to the size of context.

We compared the proposed DNN-HMM approach with the traditional HMM approach Sherman and Liu (2008) in which the emission probability is calculated from topic-dependant LMs (Sherman and Liu 2008). From Table 4, the F1-measure is improved relatively by 20% from 0.637 to 0.765 by the proposed DNN-HMM approach and the differences are significant at $p < 0.01$ (Koehn 2004).

5.4 Results of LSTM-HMM

We investigated the LSTM-HMM model to see whether explicit sequential modeling benefits the story segmentation performance. In the LSTM-HMM approach, the LSTM-RNN contains one LSTM layer followed by an output layer. The number of nodes in the output layer was the same as the number of clusters (topics). Softmax activation function was used in the output layer. There was a projection layer with 200 nodes between the input layer and the LSTM layer.

We investigated the performance of different size of memory cells and different number of clusters (topics). As shown in Table 5, the cluster number ranges from 50 to 200 and the number of memory cells on the LSTM layer ranges from 256 to 1024. The F1-measure improves when the number of memory cells increases from 256 to 768 and it begins to decrease when the number is 1024. Meanwhile, with a fixed number of memory cells, the F1-measure first increases when the cluster number ranges from 50 to 150 and then it decreases when the cluster number is larger than 150. When the cluster number is 150 and the LSTM cell number is 768, we have the highest F1-measure of 0.774, which is higher than 0.765 of the DNN-HMM approach.

5.5 Results of LSTM-MTL-HMM

As we discussed in Sect. 4.3, in the experiment, the MTL was performed by adding an additional output layer to the LSTM layer in the LSTM-RNN model. We used a trigram LM as the second task. Here the size of the secondary output layer is the same as the size of the vocabulary, which is 57,817. Adding an additional output layer apparently increases the model parameters. However, these additional parameters are just used to help

Table 2 F1-measure with different numbers of clusters for DNN-HMM

Cluster	50	100	150	170	200
F1-measure	0.719	0.725	0.742	**0.765**	0.730

Table 3 F1-measure with different size of context for DNN-HMM

Size	40	50	60	70	80
F1-measure	0.753	0.761	**0.765**	0.758	0.752

Table 4 F1-measure of the proposed DNN-HMM approach and the conventional HMM approach

Approach	F1-measure
Conventional HMM (Sherman and Liu 2008)	0.637
DNN-HMM	**0.765**

Table 5 F1-measure with different numbers of clusters and different number of nodes on LSTM layer using LSTM-HMM approach

Cluster#/nodes#	256	512	768	1024
50	0.727	0.732	0.738	0.740
100	0.739	0.747	0.758	0.761
150	0.752	0.765	**0.774**	0.770
170	0.746	0.758	0.765	0.762
200	0.737	0.748	0.756	0.751

Table 6 F1-measure with different scalar α values for LSTM-MTL-HMM

Scale	0	0.2	0.4	0.6	0.8	1.0
F1	0.774	0.775	**0.778**	0.770	0.764	0.761

Topic cluster number is set to 150 and the number of nodes for the LSTM layer is set to 768

the primary classification task in the training process and they are simply discarded when the network is well trained. Hence, the process of predicting topic posteriors and the HMM decoding are exactly the same as the previous LSTM-HMM model.

Different scalar (α) values are tested and the results are summarized in Table 6. We can see that the highest F1-measure of 0.778 is achieved when $\alpha = 0.4$, which is slightly improved as compared with the LSTM-HMM approach without MTL ($\alpha = 0$).

5.6 Comparison with state-of-the-art

We also compared the proposed approach with some state-of-the-art methods on the TDT2 corpus. The results are summarized in Table 7. We can clearly see the proposed NN-HMM approach outperforms all the methods in the comparison. The LSTM-MTL-HMM approach improves F1-measure by 6.6% relatively as compared to DD-CRP (Yang et al. 2014), a popular unsupervised story segmentation approach. The superior performances demonstrate that neural network has strong topic modeling ability and it provides a promising way for detecting story boundaries.

6 Conclusions and future work

In this paper, we have proposed a hybrid neural network hidden Markov model (NN-HMM) approach for automatic story segmentation. Specifically, we use a neural network to predict topic posterior from BOW feature vector and an HMM to model the transition between topics. The Viterbi search algorithm is used for decoding the word sequence into topic sequence, from which the story boundary can be identified when the topic changes. We have studied three different neural network architectures: a feed-forward network, an LSTM-RNN and an LSTM-RNN with the second task (a language model). LSTM-RNN has strong sequential/contextual modeling ability and multi-task learning (MTL) may benefit the generalization ability of the model. Experiments on TDT2 corpus show that the proposed approach outperforms the traditional HMM approach significantly and it achieves the state-of-art performance in story segmentation. Specifically, the LSTM-MTL-HMM approach achieves the highest F1-measure (0.778) on the TDT2 corpus. In addition, the NN predicted topic posteriors can be used as features for other story segmentation methods (e.g., DP and TextTiling) to improve story segmentation performance compared to previous features such as *tf–idf* and LDA derived features. Future research concentrates in two directions: first, we will further speed up the training process and investigate the impact of reducing output number of the second task by clustering unusual words and use the cluster labels as the training targets. Second, neural network bottleneck features (BNF) have achieved promising performance in many tasks (Zhang et al. 2014, 2015; Wu et al. 2015; Grezl et al. 2014) recently. We plan to investigate whether BNF can further boost the story segmentation performance.

Table 7 F1-measure comparison with some state-of-the-art methods

Approach	F1-measure
TextTiling (Hearst 1997)	0.553
HMM (Sherman and Liu 2008)	0.637
PLSA-DP-CE (Lu et al. 2011b)	0.682
BayesSeg (Eisenstein and Barzilay 2008)	0.710
DD-CRP (Yang et al. 2014)	0.730
DNN-HMM	0.765
LSTM-HMM	0.774
LSTM-MTL-HMM	**0.778**

Acknowledgements This paper is supported by the National Natural Science Foundation of China (61571363), Aeronautical Science Foundation of China (20155553038 and 20155553040), Science and Technology on Avionics Integration Laboratory.

References

Abdel-Hamid O, Deng L, Yu D (2013) Exploring convolutional neural network structures and optimization techniques for speech recognition. In: Proceedings of INTERSPEECH, pp 3366–3370

Banerjee S, Rudnicky AI (2006) A texttiling based approach to topic boundary detection in meetings. In: Proceedings of INTERSPEECH

Beeferman D, Berger A, Lafferty J (1999) Statistical models for text segmentation. Mach Learn 34(1–3):177–210

Blei DM, Moreno PJ (2001) Topic segmentation with an aspect hidden Markov model. In: Proceedings of SIGIR, pp 343–348

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022

Bouchekif A, Damnati G, Charlet D (2014) Intra-content term weighting for topic segmentation. In: Proceedings of ICASSP, pp 7113–7117

Bourlard HA, Morgan N (2012) Connectionist speech recognition: a hybrid approach, vol 247. Springer, Berlin

Chaisorn L, Chua TS, Lee CH (2003) A multi-modal approach to story segmentation for news video. World Wide Web Internet Web Inf Syst 6(2):187–208
Charlet D, Damnati G, Bouchekif A, Douib A (2015) Fusion of speaker and lexical information for topic segmentation: a co-segmentation approach. In: Proceedings of ICASSP, pp 5261–5265
Chen H, Guo B, Yu Z, Han Q (2016) Toward real-time and cooperative mobile visual sensing and sharing. In: Proceedings of INFOCOM, pp 1–9
Chen D, Manning CD (2014) A fast and accurate dependency parser using neural networks. In: EMNLP, pp 740–750
Chunwijitra V, Chotimongkol A, Wutiwiwatchai C (2016) A hybrid input-type recurrent neural network for LVCSR language modeling. Eurasip J Audio Speech Music Process 1:15
Collobert R, Weston J (2008) A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of ICML, pp 160–167
Cullar MP, Delgado M, Pegalajar MC (2005) An application of nonlinear programming to train recurrent neural networks in time series prediction problems. In: Proceedings of ICEIS, pp 35–42
Damavandi B, Kumar S, Shazeer N, Bruguier A (2016) Nn-grams: Unifying neural network and n-gram language models for speech recognition. In: Proceedings of INTERSPEECH, pp 3499–3503
Eisenstein J, Barzilay R (2008) Bayesian unsupervised topic segmentation. In: Proceedings of EMNLP, pp 334–343
Fiscus J, Doddington G, Garofolo J, Martin A (1999) NISTs 1998 topic detection and tracking evaluation (TDT2). In: Proceedings of the 1999 DARPA Broadcast News Workshop, pp 19–24
Fragkou P, Petridis V, Kehagias A (2004) A dynamic programming algorithm for linear text segmentation. J Intell Inf Syst 23(2):179–197
Ghosh S, Vinyals O, Strope B, Roy S, Dean T, Heck L (2016) Contextual LSTM (CLSTM) models for large scale NLP tasks. In: Proceedings of DL-KDD
Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: Proceedings of ICASSP, pp 6645–6649
Grezl F, Karafiat M, Vesely K (2014) Adaptation of multilingual stacked bottle-neck neural network structure for new language. In: Proceedings of ICASSP, pp 7654–7658
Haidar M, Kurimo M (2016) Recurrent neural network language model with incremental updated context information generated using bag-of-words representation. In: Proceedings of INTERSPEECH
Hearst MA (1997) Texttiling: segmenting text into multi-paragraph subtopic passages. Comput Linguist 23(1):33–64
Heinonen O (1998) Optimal multi-paragraph text segmentation by dynamic programming. In: Proceedings of ACL, pp 1484–1486
Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of SIGIR, pp 50–57
Huang Z, Li J, Siniscalchi SM, Chen IF, Wu J, Lee CH (2015) Rapid adaptation for deep neural networks through multi-task learning. In: Proceedings of INTERSPEECH, pp 3625–3629
James A (2002) Introduction to topic detection and tracking. Topic detection and tracking, pp 1–16
Karypis G (2002) Cluto—a clustering toolkit. Tech. Rep, DTIC Document
Kim Y (2014) Convolutional neural networks for sentence classification. In: EMNLP, pp 1746–1751
Koehn P (2004) Statistical significance tests for machine translation evaluation. In: Proceedings of EMNLP, pp 388–395
Kumar G, FD'Haro L (2015) Deep autoencoder topic model for short texts. In: Proceedings of IWES
Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Proceedings of national conference on artificial intelligence, pp 2267–2273
Larochelle H, Lauly S (2012) A neural autoregressive topic model. In: Proceedings of NIPS, pp 2717–2725
Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of ICML, pp 1188–1196
Lee L, Chen B (2005) Spoken document understanding and organization. Signal Process Mag IEEE 22(5):42–60
Li J, Cheng JH, Shi JY, Huang F (2012) Advances in computer science and information engineering. Springer, Berlin
Liu X, Gao J, He X, Deng L, Duh K, Wang YY (2015) Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In: Proceedings of HLT, pp 912–921
Li C, Wang H, Zhang Z, Sun A, Ma Z (2016) Topic modeling for short texts with auxiliary word embeddings. In: Proceedings of SIGIR, pp 165–174
Lu M, Leung CC, Xie L, Ma B, Li H (2011a) Probabilistic latent semantic analysis for broadcast news story segmentation. In: Proceedings of INTERSPEECH, pp 1109–1112
Lu M, Zheng L, Leung CC, Xie L, Ma B, Li H (2011b) Broadcast news story segmentation using probabilistic latent semantic analysis and Laplacian eigenmaps. In: Proceedings of APSIPA, pp 356–360
Malioutov I, Barzilay R (2006) Minimum cut model for spoken lecture segmentation. In: Proceedings of ACL, pp 25–32
Malioutov I, Parkand A, Barzilay R, Glass J (2007) Making sense of sound: unsupervised topic segmentation over acoustic input. In: Proceedings of ACL, p 504
Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. arXiv:1301.3781 (preprint)
Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013b) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst 26:3111–3119
Rabiner LR, Juang BH (1986) An introduction to hidden Markov models. ASSP Mag IEEE 3(1):4–16
Rau LF, Jacobs PS, Zernik U (1989) Information extraction and text summarization using linguistic knowledge acquisition. Inf Process Manag 25(4):419–428
Reynar JC (1994) An automatic method of finding topic boundaries. In: Proceedings of ACL, pp 331–333
Rosenberg A, Hirschberg J (2006) Story segmentation of broadcast news in English, Mandarin and Arabic. In: Proceedings of HLT, pp 125–128
Schultz T, Waibel A (2001) Language-independent and language-adaptive acoustic modeling for speech recognition. Speech Commun 35(1):31–51
Seltzer M, Droppo J (2013) Multi-task learning in deep neural networks for improved phoneme recognition. In: Proceedings of ICASSP, pp 6965–6969
Sherman M, Liu Y (2008) Using hidden Markov models for topic segmentation of meeting transcripts. In: Proceedings of SLT, pp 185–188
Shriberg E, Stolcke A, Hakkani-Tur D, Tür G (2000) Prosody-based automatic segmentation of speech into sentences and topics. Speech Commun 32(1–2):127–154
Soderland S (1999) Learning information extraction rules for semi-structured and free text. Mach Learn 34(1–3):233–272
Sundermeyer M, Schlter R, Ney H (2012) LSTM neural networks for language modeling. In: Proceedings of INTERSPEECH, pp 194–197
Tan T, Qian Y, Yu D, Kundu S, Lu L, Sim KC, Xiao X, Zhang Y (2016) Speaker-aware training of LSTM-RNNS for acoustic modelling. In: Proceedings of ICASSP, pp 5280–5284
Tian F, Gao B, He D, Liu TY (2016) Sentence level recurrent topic model: letting topics speak for themselves. arXiv:160402038 (preprint)

Van Mulbregt P, Carp I, Gillick L, Lowe S, Yamron J (1998) Text segmentation and topic tracking on broadcast news via a hidden Markov model approach. In: Proceedings of ICSLP

Wan L, Zhu L, Fergus R (2012) A hybrid neural network-latent topic model. In: Proceedings of AISTATS, vol 12, pp 1287–1294

Werbos PJ (1988) Generalization of backpropagation with application to a recurrent gas market model. Neural Netw 1(4):339–356

Williams RJ, Zipser D (1989) A learning algorithm for continually running fully recurrent neural networks. Neural Comput 1(2):270–280

Wu Z, Valentinibotinhao C, Watts O, King S (2015) Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In: Proceedings of ICASSP, pp 4460–4464

Xie L, Yang YL, Liu ZQ (2011) On the effectiveness of subwords for lexical cohesion based story segmentation of Chinese broadcast news. Inf Sci 181(13):2873–2891

Xie L, Zheng L, Liu Z, Zhang Y (2012) Laplacian eigenmaps for automatic story segmentation of broadcast news. Audio Speech Lang Proces IEEE Trans 20(1):276–289

Xu K, Xie L, Yao K (2016) Investigating LSTM for punctuation prediction. In: Proceedings of ISCSLP, pp 5280–5284

Yamron JP, Carp I, Gillick L, Lowe S, van Mulbregt P (1998) A hidden Markov model approach to text segmentation and event tracking. In: Proceedings of ICASSP, pp 333–336

Yang C, Xie L, Zhou X (2014) Unsupervised broadcast news story segmentation using distance dependent Chinese restaurant processes. In: Proceedings of ICASSP, pp 4062–4066

Yu D, Deng L (2015) Automatic speech recognition—a deep learning approach. Springer, Berlin

Yu D, Seltzer ML, Li J, Huang JT, Seide F (2013) Feature learning in deep neural networks-studies on speech recognition tasks. arXiv:13013605 (preprint)

Zhang Z, Wang L, Kai A, Yamada T, Li W, Iwahashi M (2015) Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification. EURASIP J Audio Speech Music Process 1:12

Zhang Y, Chuangsuwanich E, Glass JR (2014) Extracting deep neural network bottleneck features using low-rank matrix factorization. In: Proceedings of ICASSP, pp 185–189