J Sign Process Syst
https://doi.org/10.1007/s11265-017-1289-8

A Bidirectional LSTM Approach with Word Embeddings for Sentence Boundary Detection

Chenglin Xu[1,2] · **Lei Xie**[1] · **Xiong Xiao**[2]

Received: 25 April 2017 / Revised: 29 August 2017 / Accepted: 18 September 2017

Abstract Recovering sentence boundaries from speech and its transcripts is essential for readability and downstream speech and language processing tasks. In this paper, we propose to use deep recurrent neural network to detect sentence boundaries in broadcast news by modeling rich prosodic and lexical features extracted at each inter-word position. We introduce an unsupervised word embedding to represent word identity, learned from the Continuous Bag-of-Words (CBOW) model, into sentence boundary detection task as an effective feature. The word embedding contains syntactic information that is essential for this detection task. In addition, we propose another two low-dimensional word embeddings derived from a neural network that includes class and context information to represent words by supervised learning: one is extracted from the projection layer, the other one comes from the last hidden layer. Furthermore, we propose a deep bidirectional Long Short Term Memory (LSTM) based architecture with Viterbi decoding for sentence boundary detection. Under this framework, the long-range dependencies of prosodic and lexical information in temporal sequences are modeled effectively. Compared with previous state-of-the-art DNN-CRF method, the proposed LSTM approach reduces 24.8% and 9.8% relative NIST SU error in reference and recognition transcripts, respectively.

Keywords Sentence boundary detection · Word embedding · Recurrent neural network · Long short-term memory

1 Introduction

Recent years have witnessed significant progress in automatic speech recognition (ASR), especially with the development of deep learning technologies [1]. However, the output of ASR systems is typically rendered as a stream words missing of important structural information such as sentence boundaries. Below shows an example from the RT04-LDC2005T24 broadcast news corpus.[1]

ASR Output:
americans have come a long way on the tobacco road the romance is gone so joe camel smokers are out in the cold banned in baseball parks restaurants and even in some bars
Human Transcript:
Americans have come a long way on the tobacco road. The romance is gone now. So is Joe Camel. Smokers are out in the cold banned in baseball parks restaurants and even in some bars.

As we know, punctuation, in particular sentence boundaries, is crucial to human legibility [2]. Words without appropriate sentence boundaries may cause ambiguous meaning of some utterances. In a dictation system like voice

✉ Chenglin Xu
xuchenglin@ntu.edu.sg

Lei Xie
lxie@nwpu-aslp.org

Xiong Xiao
xiaoxiong@ntu.edu.sg

[1] School of Computer Science, Northwestern Polytechnical University, Xi'an, China

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

[1] https://catalog.ldc.upenn.edu/LDC2005T24.

Published online: 30 September 2017

input on mobile phones, user experience can be greatly improved if punctuations are automatically inserted as the user speaks. Besides improving readability, the presence of sentence boundaries in the ASR transcripts can help downstream language processing applications such as parsing [3], information retrieval [4], speech summarization [5], topic segmentation [6, 7] and machine translation [8, 9]. In these tasks, it is assumed that the transcripts have been already delimited into sentence-like units (SUs). Kahn et al. [3] showed that the error reduced significantly in parsing performance by using an automatic sentence boundary detection system. Matusov et al. [9] reported that sentence boundaries are extremely beneficial for machine translation. Thus, sentence boundary detection is an important precursor to bridge automatic speech recognition and downstream speech and language processing tasks.

Sentence boundary detection, also called sentence segmentation, aims to break a running audio stream into sentences or to recover the punctuations in speech recognition transcripts. This problem has been previously formulated as one of the metadata extraction (MDE) tasks in the DARPA-sponsored EARS program[2] and NIST rich transcription (RT) evaluations.[3] The goal of this work is to create an enriched speech transcript with sentence boundaries. The sentence boundary detection task is usually formulated as a binary classification or sequence tagging problem where we decide whether a candidate position should be a sentence boundary or not. The boundary candidate can be any inter-word region in a text or a salient pause in an audio stream. Features are always extracted from either text or audio stream or both near the candidate period. The features from text are named as lexical features, others from audio are called as prosodic features.

In the past several years, deep learning methods have been successfully applied to many sequential prediction and classification tasks, such as speech recognition [1, 10, 11], word segmentation [12], part-of-speech tagging and chunking [13]. A deep neural network (DNN) learns a hierarchy of nonlinear feature detectors that can capture complex statistical patterns. In a deep structure, the primitive layer in the DNN nonlinearly transforms the inputs into a higher level, resulting in a more abstract representation that better models the underlying factors of the data. Our recently proposed DNN-CRF work [14] has shown that by capturing a hierarchy of prosodic information the DNN is able to detect sentence boundary in a more effective way.

In this paper, we propose a new approach that is different from the previous work. The previous DNN-CRF approach used a DNN to capture abstract information (i.e., probabilities) on prosodic features, then integrated this information with lexical features into a CRF model. However, in this work, we capture the hierarchy of prosodic and lexical information simultaneous by using deep bidirectional LSTM model to leveraging its ability in remembering long context information. Through modeling the prosodic and lexical features at the same time, we can get some complementary and temporal information between them. Specifically, our contributions are summarized as follows:

1) We introduce three continuous valued word embeddings as new lexical features to represent word identities into the sentence boundary detection task. The first one is an unsupervised word embedding, trained by Continuous Bag-of-Words (CBOW) model [15]. The second one is derived from the projection layer of a LSTM [16] based neural network through supervised learning. The third one is extracted from the last hidden layer of the neural network. Experimental results show the word embedding is good lexical feature in the sentence boundary detection task and improves the performance significantly.
2) We propose a deep bidirectional LSTM based architecture with global Viterbi decoding for sentence boundary detection. This approach is designed to effectively utilize prosodic and lexical features, so as to exploit their temporal and complementary information. Compared with the previous DNN-CRF method, the proposed approach reduces 24.8% and 9.8% relative NIST SU Error in reference and recognition transcripts, respectively.

In Section 2, we provide a brief review on previous studies related to the sentence boundary detection task. In Section 3, we describe the proposed sentence boundary detection approach. In Section 4, the conventional prosodic and lexical features are described. After that, we introduce the new lexical features (word embedding) in Section 5. We discuss the experiments and results in Section 6. Finally, the conclusions are drawn in Section 7.

2 Related Works

For a classification or sequence tagging problem, studies mainly focus on finding useful features and models. For the sentence boundary detection task, researchers mostly investigate new features and models that are effective in discriminating sentence boundaries or non-boundaries. For the features, speech prosodic cues and lexical knowledge sources are investigated a lot. Prosodic cues, described by pause, pitch and energy characteristics extracted from the speech signals, always convey important structural information and reflect breaks in the temporal and intonational

[2] http://www.darpa.mil/iao/EARS.htm.

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

contour [17–20]. Studies show that sentence boundaries are often signaled by a significant pause and a pitch reset [6, 14, 19, 21, 22]. Lexical knowledge sources, such as Part-of-Speech (POS) tags and syntactic Chunk tags, are well known information that indicates important syntactic knowledge of sentences [21]. For the models, several discriminative and generative models have been studied, including Decision Tree (DT) [6, 22, 23], Multi-layer Perception (MLP) [24], Hidden Markov Model (HMM) [6, 21], Maximum Entropy (ME) [21], Conditional Random Fields (CRF) [14, 21, 25–27], and so on.

Inspired by the finding that the speech prosodic structure is highly related to the discourse structure [6, 28], some researchers have studied the use of only prosodic cues in sentence boundary detection. For example, Haase et al. [23] proposed a DT approach based on a set of features related to F_0 contours and energy envelopes. Shriberg and Stolcke [6] have shown that a DT model learned from prosodic features can achieve comparable performance with that learned from complicated lexical features. It is worth noting that, as compared with the lexical approaches, prosodic approaches usually do not use textual information and the influence of unavoidable speech recognition errors can be avoided. In addition, prosodic cues are known to be relevant to discourse structure across languages [29] and hence prosodic-based approaches can be directly applied to multilingual scenarios [29–31].

Although prosodic approaches have benefit in avoiding the effect of speech recognition errors, lexical information is still worth studying. Because the semantic and syntax cues are highly relevant to sentence boundaries [14, 21, 24, 32]. Stolcke and Shriberg [32] studied the relevance of several word-level features for segmentation of spontaneous speech on the Switchboard corpus. Their best results were achieved by using POS n-grams, enhanced by a couple of trigger words and biases. Similarly, on the same corpus, Gavalda et al. [24] designed a multi-layer perception (MLP) system based on the features of trigger words and POS tags in a sliding window reflecting lexical context. Stevenson and Gaizauskas [33] implemented a memory-based learning algorithm to detect sentence boundary on the Wall Street Journal (WSJ) corpus. They extracted totally 13 lexical features to predict whether an inter-word position is a boundary or not. In addition, statistical language model has been widely used in sentence boundary detection [5, 34–36] and punctuation prediction [37].

However, the above works only use either prosodic information or lexical knowledge. Good results of sentence boundary detection are often achieved by using both lexical and prosodic information, since these two knowledge sources are complementary in improving the performance. Gotoh and Renals [38] combined the probabilities from a language model and a pause duration model to make sentence boundary decisions. Later, they proposed a statistical finite state model that combines prosodic, linguistic and punctuation class features to annotate punctuation in broadcast news [39]. Kim and Woodland [40] performed punctuation insertion during speech recognition. Prosodic features together with language model probabilities were used within a decision tree framework. Shriberg et al. [6] integrated both lexical and prosodic features by a decision tree - hidden Markov model (DT-HMM) approach, where decision tree over prosodic features is followed by a hidden Markov model of lexical features. Since the HMM has a drawback that maximizes the joint probability of the observations and hidden events, as opposed to maximizing the posterior probability that would be a more suitable criterion to the classification task, Liu et al. [21] proposed a decision tree - conditional random fields (DT-CRF) approach that pushed the state-of-the-art performance of sentence boundary detection to a new level. Similar to the DT-HMM approach [6], the boundary/non-boundary posterior probabilities from the DT prosodic model were quantized and then integrated with lexical features in a linear-chain CRF. In the CRF, the conditional probability of an entire label sequence given a feature sequence is modeled with an exponential distribution. Furthermore, instead of a DT model in modeling prosodic features, our previous work [14] proposed a deep neural network - conditional random fields (DNN-CRF) approach that nonlinearly transformed the prosodic features into posterior probabilities. Then the posterior probabilities were integrated with lexical features in the way similar to the previous work [21]. This approach improved the performance a lot, because of DNN's ability in learning good representations from raw features through several nonlinear transformations.

Different from the aforementioned studies, the method developed in this paper trains the model using a rich set of both prosodic and lexical features. Besides, unlike the way of integrating different kind of features in previous DT-HMM [6], DT-CRF [21] and DNN-CRF [14] approaches, our proposed method combines the prosodic and lexical features at the beginning as the inputs of a single model without individually modeling each category features. Our motivation is to learn the salient and complementary information between the combined raw features for effectively discriminating sentence boundary or non-boundary by the model itself. Another difference is that a deep bidirectional LSTM network is used to learn effective feature representations and capture long term memory, so as to exploit the temporal information. The structure of the deep bidirectional LSTM network will circumvent the serious limitations of shallow models or DNN using a fixed window size in previous studies. Our experiments show that differences lead to significant improvement in sentence boundary detection task.

3 Proposed Deep Bidirectional LSTM Approach

The proposed sentence boundary detection approach, as shown in Fig. 1, consists of three stages: feature extraction, model training and boundary labeling. This architecture takes both prosodic and lexical knowledge sources as input features extracted in the feature extraction stage. After that, we train a deep bidirectional recurrent neural network (RNN) model based on long short-term memory (LSTM) [16] architecture (named as DBLSTM) to discover discriminative patterns from the basic features by non-linear transformations. The LSTM is well known in sequence labeling which maps the observation sequence to the class label sequence [41]. With *bidirectional* and *deep* architecture, the performance of sequence labeling can be further improved by the proposed DBLSTM approach. Finally, the global decisions are achieved over a graph using the Viterbi algorithm in the boundary labeling stage. The details of feature extraction are described in Sections 4 and 5. This section mainly describes the network architecture and the Viterbi decoding.

3.1 Definition

As mentioned before, the sentence boundary detection problem can be regarded as classification or sequence tagging problem. For a classification problem, the posterior probability $p(y_t|x_t)$ is calculated to decide which class ($y_t \in \{su, nsu\}$) should the example (t) belong to, given the input features (x_t). This probability can be the output of a neural network. For a sequence tagging problem, the most likely sentence boundary or non-boundary sequence y is

$$\begin{aligned}\hat{y} &= \arg\max_y p(y|x) \\ &= \arg\max_y p(x, y) \\ &= \arg\max_y p(y)p(x|y)\end{aligned}$$

We assume the input features are conditional independent given the events, that is

$$p(x|y) = \Pi_{t=1}^{T} p(x_t|y_t) = \Pi_{t=1}^{T} \frac{p(y_t|x_t)p(x_t)}{p(y_t)} \tag{1}$$

and the probability $p(y)$ is approximated as

$$p(y) = p(y_1)\Pi_{t=2}^{T} p(y_t|y_{t-1}) \tag{2}$$

then, the most likely sequence can thus be obtained as follows:

$$\hat{y} = \arg\max_y p(y)\Pi_{t=1}^{T} \frac{p(y_t|x_t)}{p(y_t)} \tag{3}$$

Since $p(x_t)$ is fixed and thus can be ignored in the maximization operation. In our proposed approach, the posterior probability $p(y_t|x_t)$ is the output of the neural network.

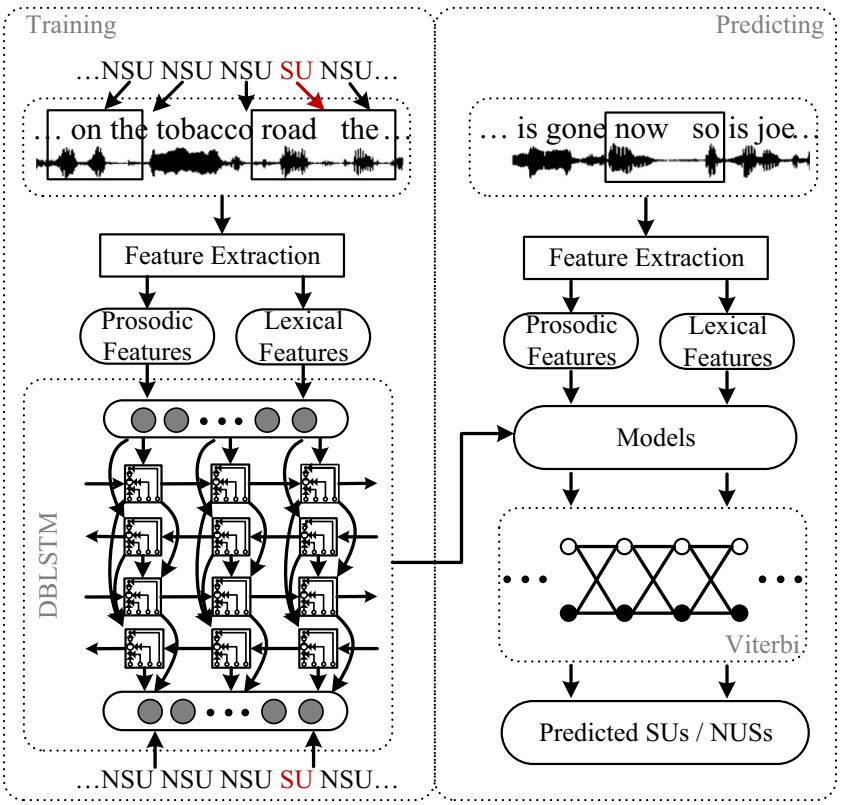


Figure 1 The architecture of our proposed sentence boundary detection system.

The following part specifies the calculation of this posterior probability.

3.2 Network Architecture

Unlike the previous DT-CRF [21] and DNN-CRF [14] frameworks of using different knowledge sources individually, the proposed approach is designed to explicitly utilize the complementary information between prosodic and lexical features by directly concatenating them together as the network's inputs. The statistical correlations among different sources can be effectively learned from the fused features by the proposed method, because the LSTM has feedbacks from previous time steps and is hence able to model temporal structure of input directly. In addition, the LSTM is able to model temporal sequences and their long range dependencies accurately. Furthermore, since context information are useful for sequential labeling task, i.e., sentence boundary detection, the deep bidirectional LSTM approach makes the decision for the input sequence by operating in both forward and backward directions to use the history and future information.

For the LSTM, the hidden state activations $\boldsymbol{h} = (h_1, \ldots, h_T)$ are iterated from $t = 1$ to T by the following equations [41, 42]:

$$i_t = f(\boldsymbol{W}_{xi}x_t + \boldsymbol{W}_{hi}h_{t-1} + \boldsymbol{W}_{ci}c_{t-1} + \boldsymbol{b}_i) \tag{4}$$

$$f_t = f(\boldsymbol{W}_{xf}x_t + \boldsymbol{W}_{hf}h_{t-1} + \boldsymbol{W}_{cf}c_{t-1} + \boldsymbol{b}_f) \tag{5}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot h(\boldsymbol{W}_{xc}x_t + \boldsymbol{W}_{hc}h_{t-1} + \boldsymbol{b}_c) \tag{6}$$

$$o_t = f(\boldsymbol{W}_{xo}x_t + \boldsymbol{W}_{ho}h_{t-1} + \boldsymbol{W}_{co}c_t + \boldsymbol{b}_o) \tag{7}$$

$$h_t = o_t \cdot g(c_t) \tag{8}$$

where i_t, f_t, o_t represent the activation values of input gate, forget gate and output gate, respectively. c_t is the state of the memory cell at time t. $f(\cdot)$ is the sigmoid activation function of the gates. $h(\cdot)$ and $g(\cdot)$ are the cell input and output activation function, respectively. $\boldsymbol{W}$ is the weight matrices, e.g., $\boldsymbol{W}_{ci}$ is the weight matrix between input gate and the memory cell. $\boldsymbol{b}$ is the bias vectors, e.g., $\boldsymbol{b}_i$ is the bias vector for input gate.

The bidirectional LSTM is proposed to use all available input information in the past and future of a specific time frame [43] by two parts: forward states, $\overrightarrow{h}_t$, and backward states, $\overleftarrow{h}_t$. The output probabilities are given as:

$$y_t = g(\boldsymbol{W}_{\overrightarrow{h}y}\overrightarrow{h}_t + \boldsymbol{W}_{\overleftarrow{h}y}\overleftarrow{h}_t + \boldsymbol{b}_y) \tag{9}$$

Finally, a DBLSTM model can be simply established by stacking multiple hidden layers of above bidirectional LSTM. Back-propagation through time (BPTT) method [44] is applied to train the model.

3.3 Tag Inference

For a sequence problem, the tag sequence should be decided globally. Given a set of tags $G = \{su, nsu\}$, we define a log transition score s_{ij} for jumping from i to j, $\{i, j\} \in G$. The valid paths of tags are encouraged, while all other paths are penalized. The score is tuned on the development set. For an input feature vector x_t, the normalized network score $p_n(y_t|x_t)$ is defined as below:

$$p_n(y_t|x_t) = log\frac{p_\theta(y_t|x_t)}{p(y_t)} \tag{10}$$

where $p_\theta(y_t|x_t)$ is the posterior probability from the network with parameter θ at input x_t, and $p(y_t)$ is the prior probability.

Given the input sequence $x_{[1:T]}$ and tag sequence $y_{[1:T]}$, we apply log operation on Eq. 3, and the whole sequence score is the sum of transition and normalized network score:

$$f(y_{[1:T]}, x_{[1:T]}, \theta) = \sum_{t=1}^{T}(s_{y_{t-1}y_t} + p_n(y_t|x_t)) \tag{11}$$

The best tag path $\hat{y}_{[1:T]}$ can be found by maximizing the sequence score:

$$\hat{y}_{[1:T]} = \arg\max_{\forall y_{[1:T]}} f(x_{[1:T]}, y_{[1:T]}, \theta) \tag{12}$$

The Viterbi algorithm is used for this tag inference.

4 Conventional Features

4.1 Conventional Lexical Features

As discussed in Section 2, syntactic tags (e.g., POS and Chunk) constitute a prominent knowledge source for sentence boundary detection. Because a sentence is usually constrained via its syntactic structure. For example, the POS tags embody syntactic information and thus can be naturally used to deduce the position of sentence boundaries. Therefore, we use POS and Chunk as syntactic features in the sentence boundary detection task. In this paper, we use the SENNA parser [13] to obtain the POS and Chunk tags given a word stream. The IOBES tagging scheme is used for chunking so as to map the word sequence to chunk stream exactly like POS. It means each word has a POS tag and a Chunk tag exactly.

4.2 Prosodic Features

In our study, as shown in Fig. 1, we consider the inter-word position as a boundary candidate and look at prosodic features of the words immediately preceding and following the candidate. A window of 200ms on both sides is also considered, as suggested in [6].

A rich set of 162 prosodic features, shown as primary cues for sentence boundary detection [6, 21, 22, 45], are collected from the audio stream at the candidate positions according to the method described in [6] and [45]. Among these features, pause and word duration features are extracted to capture prosodic continuity and boundary lengthening phenomena. Pitch and energy related features that reflect the pitch/energy declination and reset phenomena are also extracted. Since we use broadcast news as our experiment data, we also include speaker turn as a feature. From previous studies [6], speaker turn is a significant boundary cue.

5 Word Embeddings

Conventional lexical features, such as word N-grams, POS and Chunk, have shown their importance in sentence boundary detection task. However, it is not straightforward for NN to directly take these knowledge sources as their inputs. One solution is to leverage conventional one-hot representation which contains only one non-zero element in the vector with the size of the entire vocabulary. Unfortunately, such simple representation meets several challenges. One is the curse of dimensionality. Directly combination of this one-hot representation with prosodic features is not easy for a neural network to train a good model. The most critical one may be that such representation cannot reflect any relationship among different words even though they have high semantic or syntactic correlation [46]. For example, although *happy* and *happiness* have rather similar semantics, their corresponding one-hot representation vectors don't show that *happy* is much closer to *happiness* than other words like *sad*.

Recently, some complex and deep methods on learning distributed representation of word (also known as word embedding) that overcome above drawbacks have been proposed [13, 15, 47–49]. Mikolov et al. [15] proposed a continuous bag-of-words model (CBOW) for efficiently computing continuous vector representations of words from a very large unlabeled text data set. The semantically or syntactically similar words can be mapped to close positions in the continuous vector space, based on the intuition of similar words likely yielding similar context.

In this work, we firstly introduce the CBOW embedding into sentence boundary detection task as lexical features. Secondly, we propose another two supervised word embeddings to represent word identities. One is extracted from the linear projection layer of a neural network, called *projected embedding*. The other comes from the last hidden layer of the network, named as *hidden embedding*. These three words embeddings are included into lexical features.

5.1 CBOW Embedding

The CBOW model [15], as shown in Fig. 2, is similar to the feedforward neural network language model (NNLM) [47], where the hidden layer is removed and the projection layer is shared for all words. In this CBOW model, the representations of words in history and future, which comes from input layer, are summed at the projection layer followed by a hierarchical softmax [50, 51] at the output layer for computationally efficient approximation. The hierarchical softmax uses a binary tree to represent the output layer with $|V|$ words as its leaves, where $|V|$ is the vocabulary size of the entire corpus. This hierarchical softmax explicitly represents the relative probability of a leaf node conditioned on its context ($p(w_t|w_{t-c}^{t-1}, w_{t+1}^{t+c})$) by computing along the path from the root node to this leaf node using a defined energy function.[4] If there are S sequences in the data set, then the log likelihood function is as below:

$$L(\theta) = \sum_{s=1}^{S} \left(\sum_{t_s=1}^{T_s} logp\left(w_{t_s}|w_{t_s-c}^{t_s-1}, w_{t_s+1}^{t_s+c}\right) \right) \tag{13}$$

Our goal is to minimize the negative log likelihood function $f(\theta) = -L(\theta)$ through stochastic gradient descend (SGD) algorithm. Finally, continuous word embedding can be learned using this simple CBOW model.

The continuous word embedding is learned from a large of unstructured text data sets, including Wikipedia[5] and Broadcast News,[6,7] through the word2vec tool.[8] We build the CBOW model with four history and four future words at the input, by using the training criterion of correctly classifying the current (middle) word. The start learning rate is set as 0.025 by default. The threshold for occurrence of frequent words is 0.0001. Those with high frequency in the training data will be randomly down-sampled. To obtain the representation of each word appeared in the training data, the minimum count is defined as 1. At last, 100 dimensional word embeddings, which are used as proposed lexical features, are obtained through 15 iterations.

[4] In word2vec tool, the energy function is simply defined as $E(A, C) = -(A \cdot C)$, where A is the vector of a word, and C is the sum of context vectors of A. Then the probability $p(A|C) = \frac{e^{-E(A,C)}}{\sum_{v=1}^{V} e^{-E(W_v,C)}}$.

[5] http://mattmahoney.net/dc/text8.zip.

[6] https://catalog.ldc.upenn.edu/LDC2004T12.

[7] https://catalog.ldc.upenn.edu/LDC2005T24.

[8] https://code.google.com/p/word2vec/.

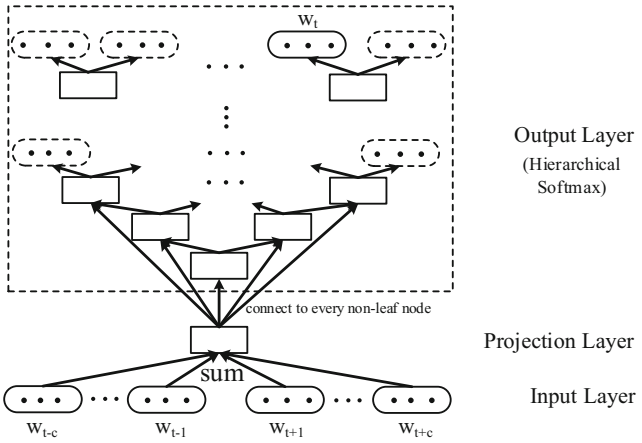


Figure 2 The CBOW model architecture includes a hierarchical softmax in output layer. Each node is represented by a vector, but only the input nodes and the leaf nodes in output layer indicate meaningful words.

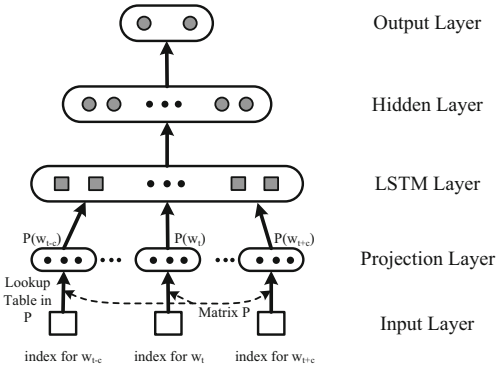


Figure 3 The architecture shows the procedure of supervised word embedding extraction. Two supervised word embeddings are extracted from projection layer and last hidden layer, respectively. For word embedding from the projection layer, only the projected vectors $(P(w_t))$ of word (w_t) at time t is used as features.

5.2 Proposed Supervised Embeddings

Different from the unsupervised word embeddings from the CBOW model, we propose another two supervised word embeddings as new lexical features for sentence boundary detection task. These two supervised vectors are extracted from a neural network learned with the supervision information of sentence boundaries, as shown in Fig. 3. The network takes several contextual words encoded by one hot representation with $|V|$ words in the vocabulary as inputs. A mapping P, being shared across all the words in the context, is applied to transform any element i of V to a low dimensional real vector $P_i \in R^m$. We name this layer as the projection layer. Given the contextual words $\{w_{t-c}^{t+c}\}$, the projected real vectors p_t associated with interested word w_t are extracted as projected embedding. These projected vectors $\{p_{t-c}^{t+c}\}$ are concatenated together to feed into the following hidden layer implemented with LSTM cells. After that, a hidden layer with sigmoid neurons are attached. The activations before applying sigmoid function in this hidden layer are extracted as the proposed hidden embedding. Finally, we attach an output layer with a softmax operation to calculate the conditional class probability.

We learn the neural network on the data sets similar to those in the CBOW model training,[9] using CNTK toolkit [52] with cross-entropy criterion. The inputs of the best network are current word with its 2 history and 2 future words. The words are encoded into one hot representations and its size is equal to the vocabulary dimension. The vocabulary, including 53,643 words, is formed by mapping words appeared less than 5 times in the data sets to unknown word. The projected word embedding size is tuned as 50, so the dimension of projection layer is 250 in total. By tuning with different numbers of nodes, the network achieves best result when the LSTM layer has 200 cells and the hidden layer has 100 nodes. The output layer has 2 nodes to calculate the posteriors of sentence boundary and non-boundary with a softmax function. The learning rate is assign as 0.01 per sample and momentum is 0.9 per mini-batch.

6 Experiments and Disscussion

6.1 Corpora and Evaluation Metrics

We evaluate the performance of sentence boundary detection using the proposed approaches on English broadcast news. Note that our approaches can be easily applied to other genres of spoken documents. The broadcast news data comes from NIST RT-04F and RT-03F MDE evaluation.[10] The released corpora from Linguistic Data Consortium (LDC) only contain the training set of the evaluations (about 40 hours). In order to keep our experimental configuration as identical as possible to [14, 21] for direct comparison, we split 2-hour data from the RT-04F released data as the testing set. Another 2-hour data is selected as the development set for parameter tuning. The rest of the data (36 hours) is used as the training set. The reference transcripts (REF) are annotated according to the annotation guideline [53], which assigns a "SU" tag at the end of a full sentence. The automatic speech recognition outputs (ASR) are generated from an in-house speech recognizer with a word error rate of 29.5%. Each inter-word position is regarded as boundary candidate. In the data, about 8% of the inter-word positions are sentence boundaries.

[9]The corresponding Wikipedia data set with sentence boundaries is used.

[10]LDC2005S16, LDC2004S08 for speech data and LDC2005T24, LDC2004T12 for reference transcriptions.

All the evaluations presented in this paper use the performance metrics including Precision, Recall, F1-measure and NIST SU error rate (SU-ER). The SU-ER, given by NIST in the EARS MDE evaluations,[11] is determined by finding the total number of inserted and deleted boundaries and dividing by the number of reference boundaries. This is the primary metric used in our comparisons. We calculate SU-ER using the official NIST evaluation tools.[12]

6.2 Experiment Setups

For the sentence boundary detection task, we train models using the aligned pairs between extracted features and boundary labels according to annotated REF transcripts, and evaluate the models on both REF and ASR transcripts. Evaluation across REF and ASR transcripts allows us to study the influence of speech recognition errors.

We first compare baseline DT and DNN methods with proposed DBLSTM method only using prosodic features. After that, our proposed new lexical features are evaluated. Finally, prosodic and lexical features are fused into the proposed DBLSTM method comparing with previous state-of-the-art DT-CRF [21] and DNN-CRF [14] approaches.

In the baseline experiments, a C4.5 decision tree is built using the WEKA toolkit[13] based on prosodic features. The DNN is fine tuned and trained by using stochastic gradient descent (SGD) on prosodic features and the minibatch includes 256 shuffled training samples. The samples are normalized so that each one is in a zero mean and unit variance distribution. To prevent overfitting, $L2$ weight decay is set to 0.00001. The learning rate is initialized as 1.0 and reduced into half when the improvement on the development set is less than 0.005. The training process will be stopped once the error on the development data starts to increase. When we integrate the posterior probabilities from DT and DNN models trained on prosodic features with lexical features into a linear-chain CRF model, we first quantize the posterior probabilities into several bins: $[0, 0.1]$, $(0.1, 0.3]$, $(0.3, 0.5]$, $(0.5, 0.7]$, $(0.7, 0.9]$, $(0.9, 1]$. Because the CRF is implemented by the CRF++ toolkit[14] and this tool can only handle discrete inputs.

We implement the DBLSTM system based on the CURRENNT tool package[15]. The conventional hidden units are replaced with the LSTM architecture in the recurrent neural network, whose objective function is a cross entropy for binary classification. The model is trained with SGD by using Back Propagation Through Time (BPTT) [44] algorithm to calculate the gradient. The network weights are initialized randomly in $[-0.08, 0.08]$ with a uniform distribution. The learning rate and momentum are 0.00005 and 0.9, respectively. We train the network with 50 parallel shuffled sequences in each epoch. To analysis the performance of each type of features, the DBLSTM model is firstly tuned with different number of hidden layers and nodes only using prosodic or lexical features. After that, a fused DBLSTM model is trained and tuned by concatenating the lexical and prosodic features into long vectors as inputs.

6.3 Experiments on Lexical Features

6.3.1 Visualization

To observe the differences among the unsupervised and supervised word embeddings, we visualize these high-dimensional data by giving each data point a location in a two-dimensional map using t-SNE tool [54]. The tool starts by converting the high-dimensional Euclidean distances between data points into conditional probabilities that represent similarities. The visualizations[16] of the three word embeddings are shown in Fig. 4. We observe that the words represented by unsupervised CBOW embedding are located symmetrically no matter whether the words are followed by sentence boundaries or not. The words located closely are similar in semantic or syntactic aspect without considering their class information. When we get the word embedding in the supervised way, the words followed by the same class (boundary or non-boundary) tend to cluster together, especially for the hidden embedding. Because the hidden embedding is much discriminative and comes from the hidden layer close to the output layer. We observe that the projected embedding shows similar picture like the CBOW embedding. The words represented by the projected embedding are also located by using class information. The projected embedding has some benefits of both CBOW and hidden embeddings.

6.3.2 Experimental Comparisons

We firstly evaluate the performance of the unsupervised and supervised word embedding features in a linear-chain CRF model, which is used to model traditional N-gram features in Liu's work [21]. The results of the N-gram and three word embeddings formed as the baseline systems are summarized in Table 1. We observe that the performances of the unsupervised CBOW embedding and supervised projected embedding are better than the conventional N-gram.

[11] http://www.itl.nist.gov/iad/mig/tests/rt/2003-fall/.

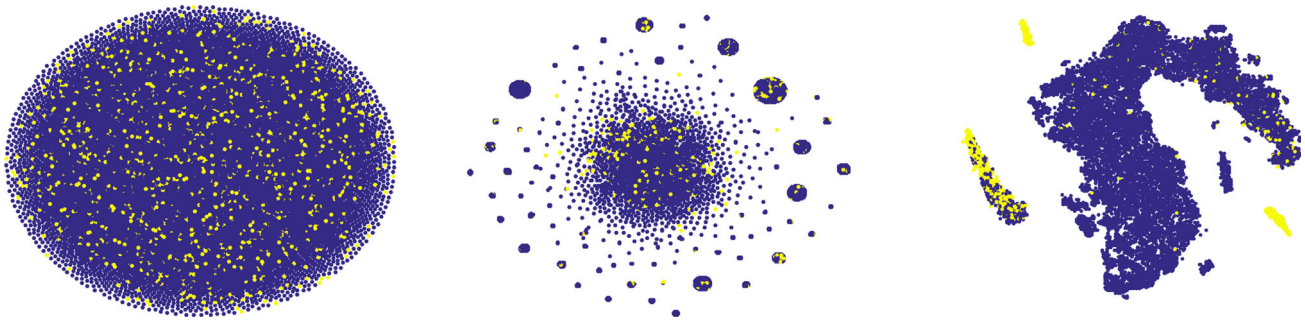
[12] See http://www.itl.nist.gov/iad/894.01/tests/rt/2004-fall/.

[13] Available at: http://www.cs.waikato.ac.nz/ml/weka/index.html.

[14] Available at: https://code.google.com/p/crfpp/.

[15] http://sourceforge.net/projects/currennt/.

[16] The initial dimension parameter of the tool is equal to each vector's size. The perplexity parameter is 50.



(a) Unsupervised CBOW embedding (b) Supervised projected embedding (c) Supervised hidden embedding

Figure 4 Visualizations of unsupervised and supervised word embeddings using t-SNE. The blue points are for non-boundaries and yellow ones are for boundaries.

However, the performance of the supervised hidden embedding is the worst in the CRF models. Although the hidden embedding becomes much discriminative, it also loses some information. Without these information, the model prefers to predict the examples as non-boundary, since the class is biased. For the CBOW and projected embeddings, the CRF model has ability to obtain the discriminative information with the supervision of class in the training stage by using a window size of 7 and a order of 2. Since the CBOW and projected embeddings contain more information than the hidden embedding, better performance are reasonable when the backend model is able to learn sequential information. Extra experiments using Support Vector Machine (SVM) show that the performance of hidden embedding is better than the CBOW and projected embeddings. Because SVM only uses the embedding of the current word and can not model the sequential information. It confirms that the hidden embedding is much discriminative.

We further evaluate the unsupervised and supervised word embedding features using our proposed DBLSTM method. Each DBLSTM model is tuned with different numbers of hidden layers and units. The best performance on the CBOW embedding is achieved when the numbers of hidden layers and units are 2 and 256, respectively. For the hidden embedding, the numbers of hidden layers and nodes are 3 and 256. For the projected embedding, the network has 3 hidden layers and 128 nodes when it achieves best performance. The results are shown in Table 1. We can see that the DBLSTM model is significantly better than the linear-chain CRF ($p < 0.05$). Another observation is that the DBLSTM models with CBOW and projected embeddings are better than the one with hidden embedding, similar to the

Table 1 The comparison among different models using lexical information.

Model	Lexical feature	REF results (%)		ASR results (%)	
		P / R / F1	SU-ER	P / R / F1	SU-ER
CRF [21]	n-gram	82.0 / 54.9 / 65.8	57.1	85.1 / 49.7 / 62.8	59.0
	n-gram,pos,chunk	81.3 / 68.9 / 74.6	47.0	80.5 / 60.3 / 68.9	52.2
CRF	CBOW embedding	79.4 / 62.7 / 70.1	53.6	83.2 / 57.4 / 67.9	54.3
	Hidden embedding	100.0 / 33.9 / 50.7	66.1	93.5 / 22.1 / 35.8	79.5
	Projected embedding	81.4 / 60.0 / 69.1	53.7	85.2 / 52.8 / 65.2	56.4
DBLSTM	CBOW embedding	81.2 / 69.2 / 74.8	46.8	81.4 / 60.5 / 69.5	53.4
	Hidden embedding	82.4 / 64.6 / 72.4	49.2	82.1 / 59.3 / 68.9	53.7
	Projected embedding	83.5 / 67.5 / 74.6	45.9	82.4 / 59.6 / 69.2	53.1
DBLSTM	CBOW, hidden embedding	80.1 / 76.8 / 75.5	45.9	81.0 / 62.6 / 70.6	52.1
	CBOW, projected embedding	83.5 / 67.7 / 74.8	45.7	82.9 / 60.0 / 69.6	52.4
DBLSTM	CBOW, hidden,pos,chunk	86.6 / 75.8 / 80.9	35.9	80.5 / 62.9 / 70.7	52.3
	CBOW, projected, pos, chunk	87.0 / 78.5 / 82.5	**33.3**	81.3 / 62.6 / 70.8	**51.8**

Results are reported by Precision (P), Recall (R), F1-measure (F1) and NIST SU Error Rate (SU-ER)

The p-test is lower than 0.05 for both REF and ASR conditions

linear-chain CRF. However, the gain becomes small. Because the DBLSTM can obtain longer contextual information from hidden embedding features using the forward and backward structures than CRF, which uses a fixed window.

Since the unsupervised and supervised word embeddings may capture different information, we integrate the unsupervised CBOW embedding with supervised projected and hidden embeddings, respectively. Since conventional syntactic information, such as POS and Chunk, are mostly used, we also combine the POS and Chunk features together to train the DBLSTM model. When we only using lexical features, the best performance is achieved by concatenating unsupervised CBOW embedding, supervised projected embedding, POS and Chunk features together. Not surprisingly, the performance on speech recognition outputs does not improve so much because of the error accumulation from wrongly recognized words in speech recognition. Specifically, the mis-recognized words make their word embedding representations inaccurate and the tags of POS and Chunk are inaccurate.

6.4 Experiments on Prosodic Features

To evaluate the DBLSTM model on prosodic features, we also tune the model with different number of hidden layers and hidden units under REF condition. The tuned model achieves the best performance when the numbers of hidden layers and hidden units are 3 and 128, respectively. With this configuration, the SU-ER reduces to 47.7%. After tuning the DBLSTM model, we further compare it with previous DT [21] and DNN [14] models. Table 2 summarizes the results of the three models in both REF and ASR test conditions. From the table, we observe that the prosodic DBLSTM model significantly outperforms the prosodic DT and DNN methods under both REF and ASR conditions (significant at $p < 0.05$ [55] for SU-ER). Specifically, when compare to DNN, the DBLSTM model achieves 8.4% and 6.5% relative reduction in NIST SU error rate for REF and ASR conditions, respectively. Because the DBLSTM has additional ability in leveraging context information in the prosodic features through hidden units with LSTM architecture, besides the DNN's ability of non-linearly transforming features into good representations. The LSTM architecture has benefits in embedding information of long time steps between relevant input and target events. We believe that the context and sequence information is essential in prosody based sentence boundary detection. For example, if the pause duration of current candidate is long and this candidate is predicted as a sentence boundary, the next candidate may have a shorter pause duration and the probability of labeling it as a boundary may become lower. Furthermore, compare with the previous DT method, the deep learning methods show its superiority in detecting sentence boundaries. As we extract a rich set of prosodic features, there must be some redundancy. The DNN and DBLSTM methods can handle these through several non-linear transforms and generate good feature representations.

Table 2 Comparison among DT, DNN and DBLSTM in experiments with prosodic features.

Transcript	Approach	Prosodic results (%)	
		P / R / F1	SU-ER
REF	DT [21]	78.8 / 56.3 / 65.7	58.8
	DNN [14]	86.9 / 56.5 / 68.5	52.1
	DBLSTM	87.7 / 60.9 / 71.9	**47.7**
ASR	DT [21]	70.6 / 56.7 / 62.9	67.0
	DNN [14]	74.3 / 61.7 / 67.4	59.7
	DBLSTM	89.3 / 50.2 / 64.3	**55.8**

Results are reported by Precision (P), Recall (R), F1-measure (F1) and NIST SU Error Rate (SU-ER)

The p-test is lower than 0.05 for both REF and ASR conditions

We also notice an apparent increase of SU error rate for all three models on ASR transcripts. This is mainly because the word errors in recognition outputs affect the prosodic feature extraction. For example, the wrong word timing information mislead the prosody extraction region, since we choose the inter-word boundary as the candidates. However, we observe that DT suffers more from the recognition errors than DNN and DBLSTM. This may indicate that neural network is more robust in processing the imperfect prosodic features.

6.5 Experiments on Combined Features

Table 3 shows the performances of lexical and prosodic information fusion. Please note that the DT-CRF and DNN-CRF systems combine the lexical features with prosodic posterior probabilities from a DT and a DNN into a linear-chain CRF, respectively. The DBLSTM model integrates the lexical and prosodic features directly at the feature level by concatenating them together as inputs. The results show that lexical and prosodic information fusion generally results in significant improvements as compared with lexical-only (Table 1) and prosodic-only (Table 2) in both REF and ASR conditions. This may indicate that prosodic information and lexical information are complimentary in sentence boundary detection.

Another observation is that the proposed DBLSTM approach outperforms both the DT-CRF and the DNN-CRF systems. In the REF condition, the DBLSTM approach decreases the SU-ER from 43.1% (DT-CRF) and 35.9% (DNN-CRF) to 27.2% with 36.9% and 24.2% relative reduction and the difference is significant ($p < 0.01$). In the

Table 3 Experimental comparisons using combined features by different fusion strategies in REF and ASR conditions.

	Approach	Information source		Results (%)	
		Lexical	Prosodic	P / R / F1	SU-ER
REF	DT-CRF [21]	n-gram, pos, chunk	DT posterior	81.4 / 73.9 / 77.4	43.1
	DNN-CRF [14]	n-gram, pos, chunk	DNN posterior	85.9 / 76.7 / 81.0	35.9
	DBLSTM	CBOW, projected, pos, chunk	prosodic features	87.4 / 85.0 / 86.2	27.2
	Viterbi	Posetriors of above DBLSTM approach		87.3 / 85.5 / 86.4	**27.0**
ASR	DT-CRF [21]	n-gram,pos,chunk	DT posterior	90.6 / 49.5 / 64.0	55.6
	DNN-CRF [14]	n-gram, pos, chunk	DNN posterior	95.0 / 49.3 / 64.9	53.3
	DBLSTM	CBOW, projected,pos,chunk	Prosodic features	79.4 / 70.1 / 74.5	**48.1**
	Viterbi	Posetriors of above DBLSTM approach		78.9 / 70.8 / 74.6	48.2

Results are reported by Precision (P), Recall (R), F1-measure (F1) and NIST SU Error Rate (SU-ER)
$p < 0.01$ in REF condition, and $p < 0.05$ in ASR condition

ASR condition, DBLSTM lowers the SU-ER to 50.9% and has 13.5% and 9.8% relative SU-ER reduction compared with DT-CRF and DNN-CRF, respectively(significant when $p < 0.05$). Comparing the precision and recall obtained from the ASR condition, the DNN-CRF system achieves better precision (95.0%) than the DT-CRF (90.6%). However, as the recall are still at a low level for both systems, the final F1 is not improved much. The proposed DBLSTM approach can greatly improve the recall rate with some drop-down of precision.

When we further employ global decisions using the posterior probabilities from the proposed DBLSTM approach by Viterbi decoding, we observe that the performance improves slightly. Compared with the predicted boundary sequences before Viterbi decoding, several non-boundaries are wrongly predicted as boundaries and some wrongly predicted as non-boundary examples are truly set as boundaries when the Viterbi algorithm is applied. The tuned transition probabilities try to increase the truly prediction of sentence boundaries and inevitably cause false alarms. The increasing number of true boundaries is slightly bigger than false alarms. The main reason of the insignificant improvement is that the DBLSTM approach is deep and bidirectional architecture, which is able to make correct decisions by optimizing contextual information from history and future. If the back-end model is weaker than the DBLSTM model, the Viterbi decoding maybe improve the performance a lot.

7 Conclusions and Future Work

We have developed a deep bidirectional recurrent neural network approach based on long short-term memory architecture in sentence boundary detection, an important speech metadata extraction (MDE) task. In addition, we introduce an unsupervised CBOW embedding containing semantic and syntactic information into sentence boundary detection task. Furthermore, we propose two supervised word embeddings (projected embedding and hidden embedding) extracted from a neural network to represent word identities. Our proposed approach has shown superior performance as compared with the previous state-of-the-art DT-CRF and DNN-CRF systems under the NIST RT-04F and RT-03F MDE evaluation framework. The improvement mainly comes from the contributions of DBLSTM in capturing long-context and complementary information from combined prosodic and lexical features. The proposed unsupervised and supervised word embeddings also contribute to the improvement.

In this work, we integrate the word embedding with prosodic features as inputs to the DBLSTM approach to predict sentence boundaries. The reason to learn word embedding is that our current paired text and speech data are not enough to build good model on the direct combination of one hot representations and prosodic features. The separately learned word embedding has benefits of using huge extra text data. In our future work, we may link the feature learning and detection model together by a flexible neural network architecture without separately learning the word embedding. The features will be automatically learned from word sequences and speech signals to reduce human efforts on feature engineering. We also notice an apparent performance degradation when we shift from reference transcripts to ASR transcripts. We will explore approaches to minimize the performance gap between clean manual transcripts and ASR transcripts with inevitable recognition errors.

References

1. Yu, D., & Deng, L. (2014). *Automatic speech recognition: a deep learning approach*. New York: Springer.

2. Jones, D.A., Wolf, F., Gibson, E., Williams, E., Fedorenko, E., Reynolds, D.A., & Zissman, M.A. (2003). Measuring the readability of automatic speech-to-text transcripts. In *INTERSPEECH*.
3. Kahn, J.G., Ostendorf, M., & Chelba, C. (2004). Parsing conversational speech using enhanced segmentation. In *Proceedings of HLT-NAACL 2004: short papers* (pp. 125–128). Association for Computational Linguistics.
4. Favre, B., Grishman, R., Hillard, D., Ji, H., Hakkani-Tur, D., & Ostendorf, M. (2008). Punctuating speech for information extraction. In *ICASSP IEEE international conference on acoustics, speech and signal processing, 2008* (pp. 5013–5016). IEEE.
5. Mrozinski, J., Whittaker, E.W., Chatain, P., & Furui, S. (2006). Automatic sentence segmentation of speech for automatic summarization. In *ICASSP 2006 proceedings ieee international conference on acoustics, speech and signal processing, 2006,* (Vol. 1 pp. I–I). IEEE (p. 2006).
6. Shriberg, E., Stolcke, A., Hakkani-Tür, D., & Tür, G. (2000). Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, *32*(1), 127–154.
7. Wang, X., Xie, L., Lu, M., CHNG, E.S., & Li, H. (2012). Broadcast news story segmentation using conditional random fields and multimodal features. *IEICE TRANSACTIONS on Information and Systems*, *95*(5), 1206–1215.
8. Xu, J., Zens, R., & Ney, H. (2005). Sentence segmentation using IBM word alignment model 1. In *Proceedings of EAMT* (pp. 280–287).
9. Matusov, E., Hillard, D., Magimai-Doss, M., Hakkani-Tür, D.Z., Ostendorf, M., & Ney, H. (2007). Improving speech translation with automatic boundary prediction. In *INTERSPEECH,* (Vol. 7 pp. 2449–2452).
10. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., & et al. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Processing Magazine*, *29*(6), 82–97.
11. Graves, A., Mohamed, A.R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 6645–6649). IEEE.
12. Zheng, X., Chen, H., & Xu, T. (2013). Deep learning for chinese word segmentation and POS tagging. In *EMNLP* (pp. 647–657).
13. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.
14. Xu, C., Xie, L., Huang, G., Xiao, X., Chng, E.S., & Li, H. (2014). A deep neural network approach for sentence boundary detection in broadcast news. In *Fifteenth annual conference of the international speech communication association*.
15. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv:1301.3781.
16. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.
17. Tseng, C., Pin, S., Lee, Y., Wang, H., & Chen, Y. (2005). Fluent speech prosody: Framework and modeling. *Speech Communication*, *46*(3), 284–309.
18. Mo, Y. (2008). Duration and intensity as perceptual cues for naïve listeners' prominence and boundary perception. In *Proceedings of the 4th speech prosody conference, Campinas, Brazil* (pp. 739–742).
19. Xie, L. (2008). Discovering salient prosodic cues and their interactions for automatic story segmentation in Mandarin broadcast news. *Multimedia Systems*, *14*(4), 237–253.
20. Mahrt, T., Cole, J., Fleck, M., & Hasegawa-Johnson, M. (2012). F0 and the perception of prominence. In *Thirteenth annual conference of the international speech communication association*.
21. Liu, Y., Shriberg, E., Stolcke, A., Hillard, D., Ostendorf, M., & Harper, M. (2006). Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Transactions on Audio, Speech, and Language Processing*, *14*(5), 1526–1540.
22. Xie, L., Xu, C., & Wang, X. (2012). Prosody-based sentence boundary detection in chinese broadcast news. In *2012 8th international symposium on chinese spoken language processing (ISCSLP)* (pp. 261–265). IEEE.
23. Haase, M., Kriechbaum, W., Möhler, G., & Stenzel, G. (2001). Deriving document structure from prosodic cues. In *Seventh European conference on speech communication and technology*.
24. Gavalda, M., & Zechner, K. (1997). High performance segmentation of spontaneous speech using part of speech and trigger word information. In *Proceedings of the fifth conference on applied natural language processing* (pp. 12–15). Association for Computational Linguistics.
25. Lu, W., & Ng, H.T. (2010). Better punctuation prediction with dynamic conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 177–186). Association for Computational Linguistics.
26. Ueffing, N., Bisani, M., & Vozila, P. (2013). Improved models for automatic punctuation prediction for spoken and written text. In *INTERSPEECH* (pp. 3097–3101).
27. Xu, C., Xie, L., & Fu, Z. (2014). Sentence boundary detection in Chinese broadcast news using conditional random fields and prosodic features. In *2014 IEEE China summit and international conference on signal and information processing (ChinaSIP)* (pp. 37—41). IEEE.
28. Hirschberg, J., & Nakatani, C.H. (1996). A prosodic analysis of discourse segments in direction-giving monologues. In *Proceedings of the 34th annual meeting on association for computational linguistics* (pp. 286–293). Association for Computational Linguistics.
29. Fung, J.G., Hakkani-Tür, D., Magimai-Doss, M., Shriberg, E., Cuendet, S., & Mirghafori, N. (2007). Cross-linguistic analysis of prosodic features for sentence segmentation. In *Eighth annual conference of the international speech communication association*.
30. Zimmerman, M., Hakkani-Tür, D., Fung, J., Mirghafori, N., Gottlieb, L., Shriberg, E., & Liu, Y. (2006). The ICSI+ multilingual sentence segmentation system. International Computer Science Inst Berkeley, CA.
31. Kolá, J., & Liu, Y. (2010). Automatic sentence boundary detection in conversational speech: a cross-lingual evaluation on English and Czech. In *2010 IEEE international conference on acoustics speech and signal processing (ICASSP)* (pp. 5258–5261). IEEE.
32. Stolcke, A., & Shriberg, E. (1996). Automatic linguistic segmentation of conversational speech. In *Proceedings of the fourth international conference on spoken language, 1996. ICSLP 96,* (Vol. 2 pp. 1005–1008). IEEE.
33. Stevenson, M., & Gaizauskas, R. (2000). Experiments on sentence boundary detection. In *Proceedings of the sixth conference on applied natural language processing* (pp. 84–89). Association for Computational Linguistics.
34. Beeferman, D., Berger, A., & Lafferty, J. (1998). Cyberpunc: a lightweight punctuation annotation system for speech. In *Proceedings of the 1998 IEEE international conference on acoustics, speech and signal processing, 1998* (Vol. 2 pp. 689–692). IEEE.
35. Mori, S. (2002). An automatic sentence boundary detector based on a structured language model. In *Seventh international conference on spoken language processing*.

36. Gravano, A., Jansche, M., & Bacchiani, M. (2009). Restoring punctuation capitalization in transcribed speech. In *IEEE international conference on acoustics, speech and signal processing, 2009. ICASSP 2009* (pp. 4741–4744). IEEE.
37. Batista, F., Moniz, H., Trancoso, I., & Mamede, N. (2012). Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts. *IEEE Transactions on Audio, Speech, and Language Processing*, *20*(2), 474–485.
38. Gotoh, Y., & Renals, S. (2000). Sentence boundary detection in broadcast speech transcripts.
39. Christensen, H., Gotoh, Y., & Renals, S. (2001). Punctuation annotation using statistical prosody models. In *ISCA tutorial and research workshop (ITRW) on prosody in speech recognition and understanding*.
40. Kim, J.-H., & Woodland, P.C. (2001). The use of prosody in a combined system for punctuation generation and speech recognition. In *Seventh European conference on speech communication and technology*.
41. Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks* Vol. 385. Heidelberg: Springer.
42. Gers, F.A., Schraudolph, N.N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, *3*(Aug), 115–143.
43. Schuster, M., & Paliwal, K.K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.
44. Williams, R.J., & Zipser, D. (1995). Gradient-based learning algorithms for recurrent networks and their computational complexity. *Backpropagation: Theory, Architectures, and Applications*, *1*, 433–486.
45. Huang, Z., Chen, L., & Harper, M. (2006). An open source prosodic feature extraction tool. In *Proceedings of the language resources and evaluation conference (LREC)*.
46. Gao, B., Bian, J., & Liu, T.-Y. (2014). Wordrep: a benchmark for research on learning word representations. arXiv:1407.1640.
47. Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*(Feb), 1137–1155.
48. Glorot, X., Bordes, A., & Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: a deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)* (pp. 513–520).
49. Tur, G., Deng, L., Hakkani-Tür, D., & He, X. (2012). Towards deeper understanding: deep convex networks for semantic utterance classification. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5045–5048). IEEE.
50. Morin, F., & Bengio, Y. (2005). Hierarchical probabilistic neural network language model. In *Aistats*, (Vol. 5 pp. 246–252).
51. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).
52. Yu, D., Eversole, A., Seltzer, M., Yao, K., Huang, Z., Guenter, B., Kuchaiev, O., Zhang, Y., Seide, F., Wang, H., & et al. (2014). An introduction to computational networks and the computational network toolkit. Microsoft Technical Report MSR-TR-2014–112.
53. Strassel, S. (2004). Simple metadata annotation specification. V6.2.
54. Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605.
55. Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *EMNLP* (pp. 388–395).

Chenglin Xu received the B.Eng. and M.Sc. degrees from Northwestern Polytechnical University (NPU), China, in 2012 and 2015, respectively. He is currently a Research Associate in Temasek Laboratories, Nanyang Technological University (NTU), Singapore. His research interests include sentence boundary detection, robust speech recognition, source separation.

Lei Xie received the Ph.D. degree in computer science from Northwestern Polytechnical University (NPU), Xi'an, China, in 2004. He is currently a Professor in the School of Computer Science in NPU. From 2001 to 2002, he was with the Department of Electronics and Information Processing, Vrije Universiteit Brussel (VUB), Brussels, Belgium, as a Visiting Scientist. From 2004 to 2006, he was a Senior Research Associate in the Center for Media Technology (RCMT), School of Creative Media, City University of Hong Kong. From 2006 to 2007, he was a Postdoctoral Fellow in the Human-Computer Communications Laboratory (HCCL), Department of Systems Engineering and information Processing, The Chinese University of Hong Kong. His current research interests include speech and language processing, multimedia and human-computer interaction. He has published more than 160 papers in major journals and proceedings, such as IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, IEEE TRANSACTIONS ON MULTIMEDIA, SIGNAL PROCESSING, PATTERN RECOGNITION, ACM Multimedia, ACL, INTERSPEECH and ICASSP.

Xiong Xiao received the B.Eng. and Ph.D. degrees in computer engineering from Nanyang Technological University (NTU), Singapore, in 2004 and 2010, respectively. He joined Temasek Laboratories, NTU in 2009 where he is now a Senior Research Scientist. His research interests include robust speech processing, spoken document retrieval, and signal processing.