TOPIC DETECTION IN CONVERSATIONAL TELEPHONE SPEECH USING CNN WITH MULTI-STREAM INPUTS

Jian Sun, Wu Guo, Zhi Chen, Yan Song

National Engineering Laboratory for Speech and Language Information Processing,
University of Science and Technology of China, Hefei, China

ABSTRACT

Topic detection for conversational telephone speech (CTS) is addressed in this paper. The low accuracy of automatic speech recognition (ASR) will cause severe performance deterioration for topic detection. To make up for this, we adopt two ASR systems, HMM-BiLSTM and CTC systems, to provide complementary information for topic detection. After obtaining two sets of different recognized transcriptions, a CNN with multi-stream inputs is trained, and the pooling layer serves as document representations. Finally, element-wise summation of document representations from two streams is used as distributed representations of the documents, which are fed into agglomerative hierarchical clustering (AHC) algorithms to obtain clustering results. The experiments on a Japanese speech corpus demonstrate that the proposed approach can significantly improve the performance of topic detection.

Index Terms— topic detection, consensus analysis, agglomerative hierarchical clustering

1. INTRODUCTION

Topic detection (TD) is a task designed for finding the set of most prominent topics in a collection of text or spoken documents, and it is a fundamental task in information management. State-of-the-art topic detection for speech includes two sub-systems. The first is a typical automatic speech recognizer (ASR), which is used to transcribe spoken utterances into text. The second sub-system is a conventional text-based TD sub-system. Both of these sub-systems have an important impact on the final TD performance.

In recent years, impressive progress has been made in the fields of both ASR and natural language processing (NLP). In terms of ASR, deep learning has replaced the Gaussian mixture model (GMM) as a mainstream acoustic modelling method. The most representative architecture is long short-term memory (LSTM) [1, 2].The LSTM combined with hidden Markov models (HMMs) has become the mainstream in the ASR field. Currently, end-to-end approaches have become popular research topics. Compared with traditional HMM/neural network systems, the end-to-end approach avoids the need for linguistic resources such as a pronunciation dictionary or phonetic context-dependency trees, which greatly simplifies the training and decoding process [3, 4, 5, 6, 7]. One of the most representative end-to-end systems is the connection temporal classification (CTC)-based framework [7].

For the topic detection task, documents are first represented as fixed-dimensional feature vectors, and then clustering algorithms, such as the agglomerative hierarchical clustering (AHC) algorithm [8], are performed to partition the documents into groups. The most widely used approach for document representation is term frequency-inverse document frequency (TF-IDF) [9]. Furthermore, generative statistical models are proposed to capture the latent semantic structure of documents. Typical methods include latent semantic analysis (LSA) [10], probabilistic latent semantic analysis (PLSA) [11] and latent Dirichlet allocation (LDA) [12]. Neural network-based methods, such as neural autoregressive density estimators (DocNADE) [13], have also been investigated for document representations. Moreover, convolutional neural networks (CNNs) [14, 15] are employed to capture n-gram features and construct document representations. However, topic detection is essentially an unsupervised task, and the lack of predefined labels is a problem for topic detection. To enhance the quality of pseudo-labels for model training, Chen et al. used consensus analysis to select training samples [16].

In this paper, we investigate the topic detection task on conversational telephone speech (CTS). Since topic detection relies fundamentally on matching words and phrases among different transcript documents, the accuracy of TD will be inevitably affected by speech recognition errors. Different ASR systems can provide complementary information for topic detection. Word-level transcription can provide more semantic information, but it cannot solve the out-of-vocabulary (OOV) problem. Some OOVs contain very important discriminative topic information. On the other hand, the grapheme system can partly solve the OOV problem. In this work, we take two

This work was partially funded by the National Key Research and Development Program of China (Grant No. 2016YFB1001303) and the National Natural Science Foundation of China (Grant No. U1836219).

ASR systems, based on words and graphemes, to transcribe speech. After obtaining two sets of transcriptions, a CNN with multi-stream inputs is trained, and the pooling layers of the trained CNN serve as a distributed representation of the documents. Finally, we use the AHC algorithm on the document representations. Experiments are carried out on a spontaneous Japanese CTS corpus, and the results demonstrate the effectiveness of the proposed method. The contribution of this paper is as follows.

1. We propose a CNN to integrate two different ASR recognized results. The proposed CNN has two different sets of input nodes: the embedding matrix of the words and the embedding matrix of the graphemes. In the training process, instead of concatenating the embedding of different recognized tokens as CNN inputs, we feed the CNN with them separately.

2. We use the average-pooling layer before the output nodes as a document representation. We also separately feed the CNN with the embedding of different recognized tokens to extract two sets of document representations. Instead of concatenating these two sets of document representations, we use the elementwise summation of two vectors to obtain the final distributed representation of documents for topic detection.

The rest of this paper is organized as follows: Section 2 provides a detailed description of the topic detection system, especially the multi-stream training framework. Then, Section 3 presents our experimental setup and other details, including the experimental results. Finally, the discussion and conclusion are presented in Section 4.

2. PROPOSED METHODS

In this section, we will discuss the proposed multi-stream CNN framework, especially the training and document representation extraction procedures. The overall architecture is depicted in Fig.1. The process can be divided into the following three steps.

2.1. ASR

As mentioned above, we adopt two totally different ASR systems to provide complementary information. The first adopts HMM-BiLSTM as an acoustic model and tied tri-phones as modelling units; a 3-gram word-based language model is used in the decoding process. The second adopts the CTC acoustic model and graphemes as modelling units; a 3-gram grapheme-based language model is used in the decoding process.

2.2. Consensus analysis for coarse label generation

Topic detection has no prior label information and is intrinsically an unsupervised clustering task. We use consensus analysis to generate pseudo-labels and select documents with

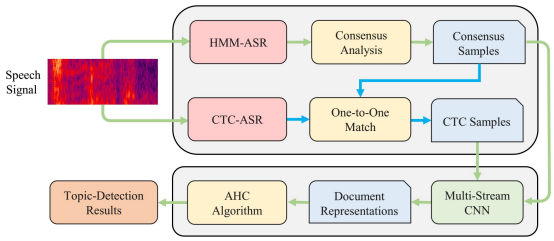

Fig. 1: Architecture of the proposed model.

high confidence for the CNN training [16]. In the consensus analysis, we use the recognized documents of the HMM system, which provides more reliable topic information than the CTC system. All recognized documents are first converted into low-dimensional vectors through unsupervised methods, such as LDA, LSA or DocNADE. Then, we adopt the AHC algorithm to generate cluster labels for all documents. Since consensus analysis is used, two sets of vectors (such as LDA and DocNADE) are adopted to obtain two different clustering labels $\mathbb{C}1$ and $\mathbb{C}2$. With the obtained clustering labels $\mathbb{C}1$ and $\mathbb{C}2$, consensus analysis is employed on $\mathbb{C}1$ and $\mathbb{C}2$ to select consensus samples. First, a mapping function is employed to map each cluster label in $\mathbb{C}1$ to the best-matched cluster label in $\mathbb{C}2$. Second, for a spoken document d_i , if the clustering label in $\mathbb{C}1$ is equal to the after-mapping cluster label in $\mathbb{C}2$, d_i belongs to the consensus sample set; otherwise, d_i will not be used for CNN training. A detailed description of consensus analysis is available in [16].

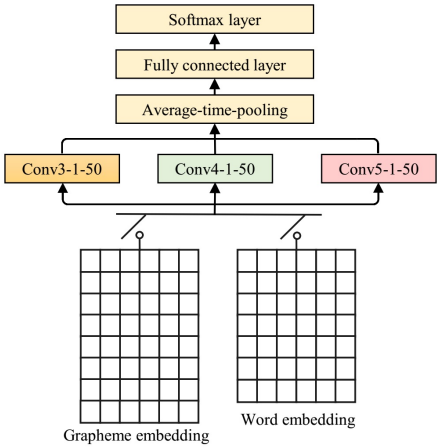

Fig. 2: Architecture of the multi-stream training framework. In this paper, we perform the convolutions over the embedded vectors using multiple filter sizes, for example, sliding over 3, 4 or 5 words (or graphemes) at a time.

2.3. CNN with multi-stream inputs

As depicted in Fig.2, the CNN architecture in this work consists of five parts: an input embedding layer, a convolutional layer, a pooling layer, a fully connected layer and a softmax layer. The major difference between our model and conven-

tional CNNs is the input layers. The proposed model has two sets of different inputs: one is used for embedding the matrix for words, and the other is used for graphemes. The output layer is the pseudo-supervised labels generated by consensus analysis in Section 2.2. All other layers are the same as conventional CNNs.

Let $\mathbf{W}_i = (w_1^i, w_2^i, ..., w_P^i)$ and $\mathbf{G}_i = (g_1^i, g_2^i, ..., g_Q^i)$ represent the recognized results of the HMM and CTC systems, respectively, for a training document d_i. We first map the results to embedding matrices, $\mathbf{M}^{W_i}$ and $\mathbf{M}^{G_i}$ [17]. Instead of concatenating the embedding matrices of $\mathbf{M}^{W_i}$ and $\mathbf{M}^{G_i}$, we feed them into the CNN separately in the training procedure. This means that the training samples can be doubled and the dimensions of $\mathbf{M}^{W_i}$ and $\mathbf{M}^{G_i}$ can be different. For example, in the first iteration, we choose the word embedding input of mini-batch training data $\mathbf{M}^W$, and in the second iteration, grapheme embedding input $\mathbf{M}^G$ will be utilized. Convolution operation is applied to the embedding matrix:

$$\boldsymbol{c} = f(\mathbf{W}_{conv} * \mathbf{M} + b) \tag{1}$$

where $\mathbf{W}_{conv} \in \mathbb{R}^{n \times h \times k}$ are the convolution weights, $\mathbf{M}$ can be either $\mathbf{M}^W$ or $\mathbf{M}^G$, b is the bias and f is a non-linear function. n is the number of feature maps, h is the filter width, and k is the dimension of word vectors and grapheme vectors. We then apply an average-over-time pooling operation over $\boldsymbol{c}$, so both word and grapheme documents can be converted to the same dimensional vectors after the pooling layer. When the training data are from the word embedding branch, the loss of the mini-batch data is calculated as:

$$O = -\log P(y_{label}|\mathbf{W}) \tag{2}$$

where $P(y_{label}|\mathbf{W})$ is the predicted distribution. Compared with equation (2), the objective of the grapheme embedding branch is presented as equation (3), with an adjustable parameter α, where $0 \leqslant \alpha \leqslant 1$.

$$O = -\alpha \log P(y_{label}|\mathbf{G}) \tag{3}$$

With the trained model, we can extract the document representations of the document d_i, which are the output of the pooling layer. Since there are two inputs, the word and grapheme embedding matrices, each input can obtain one vector, denoted as $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Then, we apply an element-wise summation of the two vectors to obtain the final distributed representation of one document.

$$\boldsymbol{v} = \boldsymbol{v}_1 \oplus \boldsymbol{v}_2 \tag{4}$$

where $\oplus$ is the element-wise summation operator. After obtaining the distributed representation of all documents, the AHC algorithm can be applied to categorize the documents into different topics.

3. EXPERIMENT

In this section, we describe the datasets and the topic detection performance of the proposed approach.

3.1. Database

The experiments are conducted on two Japanese CTS corpora collected by the Speechocean Corporation, King-ASR-222-2 and King-ASR-222-3. The King-ASR-222-2 corpus is used to build the ASR systems, while all the topic detection experiments are conducted on the King-ASR-222-3 corpus.The King-ASR-222-2 corpus contains 120-hour spontaneous dialog speech data. In the experiments, we randomly selected 105 hours and 12 hours of speech data as the training and cross-validation sets. The King-ASR-222-3 corpus consists of 7435 recorded telephone conversations with approximately 200 hours of speech in total. This corpus contains 21 topics, and each conversation is assigned to a specific topic. We use the King-ASR-222-3 corpus as the ASR test set, and the recognized results are fed into the topic detection modules.

3.2. ASR modules

The Kaldi [18] and Eesen [19] toolkits are used in the HMM and CTC ASR systems, respectively. Both systems adopt BiLSTM networks, which contain 3 layers with 1024 nodes in each layer. The acoustic feature is 108-dimensional filter-bank features (36 filter-bank features, delta coefficients, and delta-delta coefficients) with mean and variance normalization. For the HMM system, tied tri-phones (senones) are selected as acoustic model units. There are 237 monophones and 13566 senones in acoustic modelling, and a 3-gram word-based language model is used in the decoding procedure. For the CTC system, we take 2488 different graphemes (hiragana, katakana and kanji) plus blank as 2489 output nodes in the acoustic model. A 3-gram grapheme-based language model is used in the decoding procedure. Word error rate (WER) and character error rate (CER) are used as evaluation criteria for the HMM and CTC systems, respectively. Table 1 shows the experimental results of the related systems.

From Table 1, it can be seen that the accuracies of both of these systems are not very satisfactory. The reason may be the low signal-to-noise ratio (SNR), channel mismatch and spontaneous speaking style.

3.3. Topic detection

With the HMM- (word documents) and CTC-recognized results (grapheme documents), topic detection experiments are conducted on the King-ASR-222-3 corpus. The number of clusters is set to 21 in our experiments, and we evaluate the clustering performance by two metrics, the accuracy (ACC) and the normalized mutual information (NMI) [20].

3.3.1. Unsupervised models

We first conduct the clustering algorithm on the document representations inferred by conventional unsupervised methods. Three methods, LSA, LDA and DocNADE, are com-

Table 1: The ASR systems with WER or CER on King-ASR-222-3 corpus

ASR system	WER%(CER%)
HMM-BiLSTM	44.91%
CTC-BiLSTM	45.92%

Table 2: The performance of the unsupervised model

Model	word		grapheme	
	ACC	NMI	ACC	NMI
LDA	0.4260	0.3656	0.2779	0.2111
LSA	0.4296	**0.3767**	**0.3198**	**0.2459**
DocNADE	**0.4576**	0.3732	0.2667	0.1952

pared.In LSA, we retain the top 60 singular values to form the new subspace for both word documents and grapheme documents. In LDA, the number of latent topics is set to 60 and 40 for word and grapheme documents, respectively. In DocNADE, the hidden layer size is set to 40, and the sigmoid is chosen as the activation function for both word and grapheme documents.

The performances of the related systems are shown in Table 2. The performance based on the grapheme documents is obviously worse than that based on the word documents, which means word-based systems retain much more semantic information from the documents.

3.3.2. CNNs trained with consensus analysis samples

There are 7435 documents in the King-ASR-222-3 corpus. After consensus analysis, the number of documents for LDA-DocNADE and LSA-DocNADE is 3296 and 3215 respectively. Consensus analysis only selects the samples with high confidence for CNN training.

We implement the CNNs by TensorFlow [21]. The input feature lengths for CNNs are set to 270 for word documents and 700 for grapheme documents. In the convolution layer, different filter windows of 3, 4, and 5 with 50 feature maps each are adopted, and we also use a dropout rate of 0.2 before the fully connected layer. The word and grapheme-embedding vectors are initialized independently with 300-dimensional pre-trained word2vec vectors. All of the pre-trained word and grapheme vectors are updated along with other model parameters by Adam during the CNN training [22]. The last pooling layer of the trained CNN is used as a document representation. Depending on the inputs, each document can extract two vectors (word and grapheme vectors). These two vectors can be added using equation (4) (denoted Vector-A), and they can also be concatenated (denoted Vector-C) to form a high vector.

Table 3 shows topic detection based on these four types of vectors. The CNN system can achieve noticeably better performance than the unsupervised systems in Table 2. Furthermore, the Vector-A system can achieve the best performance,

Table 3: The performance of multi-stream inputs

Model	LDA-DocNADE		LSA-DocNADE	
	ACC	NMI	ACC	NMI
Word	0.5087	0.4340	0.5029	0.4342
Grapheme	0.3806	0.3431	0.4300	0.3555
Vector-C	0.5243	0.4631	0.5190	0.4623
Vector-A	**0.5712**	**0.5245**	**0.5578**	**0.5158**

Table 4: The performance of single-stream inputs

Model	LDA-DocNADE		LSA-DocNADE	
	ACC	NMI	ACC	NMI
W-CNN	0.4909	0.4348	**0.4912**	0.4209
G-CNN	0.2449	0.1882	0.2616	0.1926
Vector-WG	**0.4929**	**0.4399**	0.4897	**0.4428**

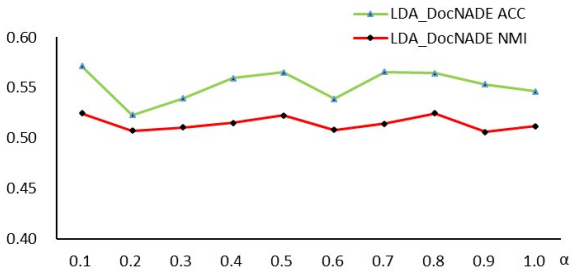

Fig. 3: The topic detection performance of the Vector-A system with different α.

with 6.25% absolute improvements for ACC and 9% for NMI.

In the experiments, the α in equation (3) controls the impact of the grapheme documents on model training. When α is set from 0.1 to 1.0 with an interval of 0.1, the topic detection performances of the Vector-A system are listed in Fig.3. As presented in Fig.3, performance varies with different weights α, especially for ACC. The system obtains the best performance for both ACC and NMI with an α of 0.1.

To test the effectiveness of the proposed method, we build two contrastive CNNs with only a single-stream input, denoted W-CNN and G-CNN, depending on the word or grapheme document input. All other configurations are the same as multi-stream systems. Additionally, the concatenation of the vectors (denoted Vector-WG) extracted from W-CNN and G-CNN is applied to improve the performance. Comparing Table 4 with Table 3, the performances of contrastive CNNs are not satisfactory.

4. CONCLUSION

We propose a multi-stream CNN training framework to fuse the word and grapheme recognized transcriptions in this paper. Despite different inputs, high-level parameters can be shared in model training and final document representation extraction. The experimental results demonstrate the effectiveness of the proposed framework.

5. REFERENCES

[1] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[2] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[3] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[4] Rohit Prabhavalkar, Tara N Sainath, Bo Li, Kanishka Rao, and Navdeep Jaitly, "An analysis of "attention" in sequence-to-sequence models,"," in *Proc. of Interspeech*, 2017.

[5] Kartik Audhkhasi, Bhuvana Ramabhadran, George Saon, Michael Picheny, and David Nahamoo, "Direct acoustics-to-word models for english conversational speech recognition," *arXiv preprint arXiv:1703.07754*, 2017.

[6] Rohit Prabhavalkar, Kanishka Rao, Tara N Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proc. Interspeech*, 2017, pp. 939–943.

[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[8] Lior Rokach and Oded Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*, pp. 321–352. Springer, 2005.

[9] Gerard Salton and Christopher Buckley, "Term-weighting approaches in automatic text retrieval," *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[10] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[11] Thomas Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine learning*, vol. 42, no. 1-2, pp. 177–196, 2001.

[12] David M Blei, Andrew Y Ng, and Michael I Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[13] Hugo Larochelle and Stanislas Lauly, "A neural autoregressive topic model," in *Advances in Neural Information Processing Systems*, 2012, pp. 2708–2716.

[14] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[15] Yoon Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[16] Peixin Chen, Wu Guo, Lirong Dai, and Zhenhua Ling, "Pseudo-supervised approach for text clustering based on consensus analysis," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6184–6188.

[17] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[18] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[19] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 167–174.

[20] Junyuan Xie, Ross Girshick, and Ali Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.

[21] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al., "Tensorflow: a system for large-scale machine learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[22] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.