Exploiting end of sentences and speaker alternations in language modeling for multiparty conversations

Hiroto Ashikawa*, Naohiro Tawara*, Atsunori Ogawa†,
Tomoharu Iwata†, Tetsunori Kobayashi*, Tetsuji Ogawa*
* Department of Communications and Computer Engineering, Waseda University, Japan
† NTT Communication Science Laboratories, NTT Corporation, Japan

***Abstract*—The effective handling of end-of-sentences and speaker alternations, both of which are frequently observed in multiparty conversations, in recurrent neural network language models (RNNLMs) is investigated. This kind of auxiliary information is represented as context cues and feature vectors. The former representation can be inserted directory into a transcription and treated as a word token, while the latter serves as auxiliary input to the neural networks. Experimental comparisons using multiparty conversation data, including the AMI meeting corpus, demonstrated that both representations contribute to improvement of the RNNLMs, and that dealing with the end-of-sentences is important, especially on the multiparty conversations.**

I. Introduction

Deep learning and related technologies have improved the performance of speech recognition systems [1]. Recognizing speech in multiparty conversations, however, has not yet achieved practical levels of performance because challenges specific to this domain remain unresolved [2], [3], [4]. For example, speakers in multiparty conversations are dynamically alternated, and very short utterances that are attributed to back-channel feedback and nods frequently appear, which yields frequent end-of-sentences. Since different speakers indicate different choices of spoken words (i.e., speaking styles), subsequent words can be influenced by speakers, even if the flows of conversation are consistent. Therefore, incorporating information that is specific to multiparty conversations (e.g., dialog states and speaker alternations) into language modeling can contribute to improvements in predicting words spoken during multiparty conversations.

Many efforts have been made to develop language models for spoken dialogs. The content and subjects of dialogs have been focused on in the field of language modeling [5], [6]. Several attempts have been made to consider speakers and their alternations in language modeling, i.e., roles of participants in a meeting were identified [7] and exploited as prior information [8], [9], the effect of speakers was considered using topic models [10], [11], [12], speaker alternations were explicitly modeled [13], [14], and utterances just before alternation were explicitly considered to model interactions [15], [16]. All these methods, however, were implemented using n-gram language models.

The present study focused on end-of-sentences and speaker alternations, both of which frequently appear in multiparty conversations, and we explored the effective use of these kinds of information in recurrent neural network language models (RNNLMs) [17], including long short-term memory language models (LSTMLMs) [18], which have become the mainstream of language modeling. Two attempts have particularly been made to incorporate information on end-of-sentences and speaker alternations into RNNLMs and evaluated on multiparty conversation data including the augmented multiparty interaction (AMI) meeting corpus. The first attempted to represent these kinds of information as context cues, and inserted them directly into a transcription and treated them as word tokens. Second, they were represented as feature vectors and taken as auxiliary inputs to neural networks in addition to corresponding word vectors. The knowledge obtained in the present study could contribute to improvements in the performance of automatic speech recognition (ASR) systems and natural language processing (NLP) applications for multiparty conversations.

The rest of the present paper is organized as follows. Section II briefly reviews RNNLMs. Section III describes the use of the end-of-sentences and speaker alternations in RNNLMs. Section IV investigates the effectiveness of using these kinds of information on the performance of RNNLMs using transcriptions from Japanese and English multiparty conversations. Finally, a summary is presented in Section V.

II. RNNLMs

An RNNLM is a language model based on a two-layer neural network with an input layer, $\mathbf{x}(t)$, a hidden layer, $\mathbf{h}(t)$, and an output (word prediction) layer, $\mathbf{y}(t)$. The hidden layer has recurrent connections, which makes it possible to propagate contextual information.

Assume that the word vector at time t, denoted as $\mathbf{w}(t)$, is represented by 1-of-K encoding. Then, input $\mathbf{x}(t)$ is formed by concatenating $\mathbf{w}(t)$ and previous hidden layer output $\mathbf{h}(t-1)$ as:

$$\mathbf{x}(t) = \Big[\mathbf{w}(t)^{\mathrm{T}}\mathbf{h}(t-1)^{\mathrm{T}}\Big]^{\mathrm{T}}. \tag{1}$$

The $\mathbf{x}(t)$ is mapped to a continuous vector, $\mathbf{h}(t)$. Considering $\mathbf{h}(t)$ as a context, the network finally yields word probability distribution $\mathbf{y}(t)$, which predicts the subsequent word (i.e., the word at time $t+1$) given the context. The $\mathbf{h}(t)$ and $\mathbf{y}(t)$ are calculated as:

$$\mathbf{h}(t) = f\Big(\mathbf{U}\mathbf{x}(t)\Big),\ \mathbf{y}(t) = g\Big(\mathbf{V}\mathbf{h}(t)\Big), \tag{2}$$

where $\mathbf{U} = (u_{ji})$ denotes the weight between $\mathbf{h}(t)$ and $\mathbf{x}(t)$, and $\mathbf{V} = (v_{kj})$ denotes the weight between $\mathbf{y}(t)$ and $\mathbf{h}(t)$. The terms, $f(z)$ and $g(z)$, represent the activation function. Back propagation through time (BPTT) [19] is exploited for training.

Long short-term memory language models (LSTMLMs) [18] were also exploited in the present study to capture contexts that were longer than those that the RNNLMs could deal with. The LSTM cells, which have an error carousel and three gates, viz.,the input gate, forget gate, and output gate, are introduced into a hidden layer in this model.

III. Use of End-of-Sentences and Speaker Alternations in RNNLM

This section describes the importance of exploiting end-of-sentences and speaker alternations in language modeling for multiparty conversations. In addition, two methods are presented to incorporate the previous information into RNNLMs: first, the information is represented by the context cues and equivalently inserted into the transcriptions as other word tokens, and second, the information is represented by the auxiliary features and taken as inputs to the neural networks.

A. Phenomena specific to multiparty conversations

Effective handling of the phenomena that are frequently observed in multiparty conversations, such as end-of-sentences and speaker alternations, can enhance the language models for multiparty conversations. In contrast to spoken lectures, which are uttered by a single speaker, multiparty conversations have two main characteristics:

- Very short utterances such as back-channel feedback and nods to other utterances frequently appear and
- Directions and content of subsequent conversations depend on who is speaking.

The former property indicates that end-of-sentences frequently appear in the text transcribed from multiparty conversations. Therefore, the language models for multiparty conversations are required to capture end-of-sentences and deal with very short utterances more accurately than those in spoken lectures. In addition, the effective use of speaker-derived information could contribute to improvements to performance in predicting words in multiparty conversations because of the latter property.

B. Context cue representation

Context cues are tokens that represent arbitrary information and are inserted into transcriptions. End-of-sentences are generally represented as context cues and exploited. However, the present study not only attempted to deal with end-of-sentences but also speaker alternations using the representation of context cues. The end-of-sentences and presence or absence of speaker alternations in this case are represented by the context cues listed in Table I, and the proposed context cues are included in the transcriptions. Figure 1 has an example of using these context cues. This figure indicates that information on the end-of-sentences and speaker alternations is equivalently given to neural networks as other word tokens.

TABLE I
Context cue representation of end-of-sentences and speaker alternations. "Yes" and "No" correspond to presence and absence of end-of-sentences and speaker alternations at next utterance.

context cue	end-of-sentence	speaker alternation
$< /s >$	Yes	-
$< /s + turn >$	Yes	Yes
$< /s + else >$	Yes	No

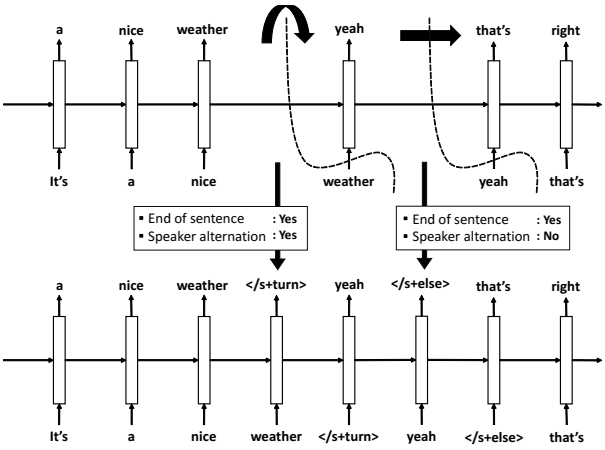

Fig. 1. Example of exploiting information on end-of-sentences and speaker alternations as context cues in RNNLMs. Upper figure corresponds to unrolled RNN that does not deal with these context cues and lower figure corresponds to that dealing with these context cues.

C. Feature vector representations

An auxiliary feature vector, $\mathbf{s}(t)$, which represents end-of-sentences and speaker alternation, for word w_t spoken at time t is defined using 1-of-K representation, as summarized in Table II. The auxiliary feature is taken as the input to the neural network at every time step, as outlined in Fig. 2. The use of the auxiliary feature and network structure are determined from preliminary experiments. The auxiliary feature at the next time, $\mathbf{s}(t+1)$, in this model is fed as bias into the hidden layer. The history of words and auxiliary information is retained in this case via the hidden layer with recurrent connections. The outputs of the word prediction layer are given in Eq. 2, as with the conventional RNNLM. The hidden layer outputs can be calculated as:

$$\mathbf{h}(t) = f\left(\mathbf{U}\left[\mathbf{w}(t)^{\mathrm{T}}\mathbf{s}(t+1)^{\mathrm{T}}\mathbf{h}(t-1)^{\mathrm{T}}\right]^{\mathrm{T}}\right). \quad (3)$$

TABLE II
Definition of auxiliary feature vectors that represent end-of-sentences and speaker alternation. Auxiliary feature $\mathbf{s}(t)$ is defined on basis of whether word w_t at time t is beginning of utterance and whether speaker alternation occurs at t.

auxiliary feature	beginning-of-sentence	speaker alternation
$(1, 0)$	Yes	-
$(0, 1)$	No	-
$(1, 0, 0)$	Yes	Yes
$(0, 1, 0)$	Yes	No
$(0, 0, 1)$	No	-

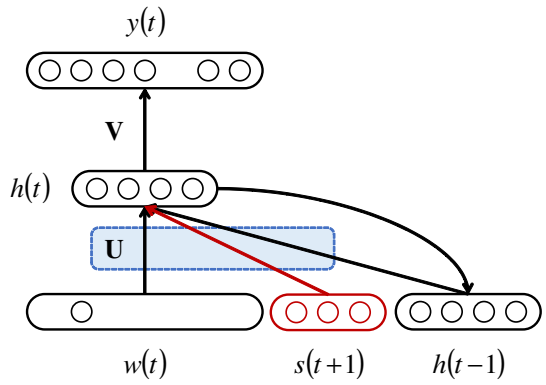

Fig. 2. RNNLM with speaker-aware training.

IV. Experiments

Experimental comparisons were conducted on multiparty conversational data to evaluate the effectiveness of exploiting information on the end-of-sentences and speaker alternations in RNNLMs in terms of test set perplexity and word error rate that were yielded from speech recognition systems. Six language models were evaluated. After this, "**C**" means context cues, "**E**" means end-of-sentences, and "**A**" means speaker alternations.

- **RNN**: RNNLM (LSTMLM) that neither deals with end-of-sentences nor speaker alternations,
- **RNN_CE**: RNNLM (LSTMLM) that deals with end-of-sentences represented by context cues,
- **RNN_E**: RNNLM (LSTMLM) that deals with end-of-sentences represented by auxiliary feature vectors,
- **RNN_CE_E**: RNNLM (LSTMLM) that deals with end-of-sentences represented by both context cues and auxiliary feature vectors,
- **RNN_CEA**: RNNLM (LSTMLM) that deals with end-of-sentences and speaker alternations represented by context cues, and
- **RNN_EA**: RNNLM (LSTMLM) that deals with end-of-sentences and speaker alternations represented by auxiliary feature vectors.

The context cues and auxiliary feature vectors in the present experiments are defined in Table I for the former and Table II for the latter.

A. Transcriptions

The transcriptions of multiparty conversations in Japanese and those in English were used for evaluations as follows.

1) NTT meeting corpus: Multiparty conversations on specific topics were collected and transcribed in the NTT Communication Science Laboratory. The transcriptions are referred to as the "NTT corpus", where all conversations were in Japanese. Four to six participants had discussions on specific topics for about 16 minutes, which was referred to as "one session." One participant was the presenter for each session and the remaining participants asked questions. Several parties participated in more than one session. Table III lists the numbers of words, utterances, sessions, and vocabulary size for the training, development, and testing data.

2) AMI meeting corpus: The AMI meeting corpus [20] includes transcriptions of multiparty meetings in which all the utterances are spoken in English. Table IV lists the recording times and numbers of words, utterances, and vocabulary size for the training, development, and testing data.

B. Experimental setup

RNNLMs were applied to evaluation on the NTT corpus because this contained relatively small amounts of data, while LSTMLMs were used in evaluation on the AMI corpus because large amounts of data were available. The parameters for RNNLMs and LSTMLMs are listed in Table V for the former and Table VI for the latter. The learning rate for both models was initialized to 0.1 and then halved when the logarithmic likelihood ratio on the development data was less than 1.003. Dropout [21] was applied in training the LSTMLMs. The emission probabilities of context cues were ignored, i.e., the corresponding probabilities were forced to zero and those of other words were normalized for cases in which the context cues were used for representing auxiliary information on the end-of-sentences and speaker alternations.

C. Experimental results

Table VII lists the test set perplexities obtained from the language models evaluated on the NTT meeting corpus and Table VIII lists those evaluated on the AMI meeting corpus.

Tables IX and X list the word error rates that were yielded from the speech recognition systems using the NTT corpus for the former and the AMI corpus for the latter. A 100-best list was generated for each utterance using a weighted finite-state transducer (WFST)-based speech recognizer [22], [23], where the recognizer utilized a fully-connected deep neural network (DNN) acoustic model and a 3-gram language model, which were trained on the previously given dialogue data. The DNN acoustic model in the experiment on the NTT corpus had six hidden layers of 2048 units, an input layer of 418 nodes, and 3874 output units. The DNN acoustic model when the AMI corpus was used had seven hidden layers of 450 units, an input layer of 220 units, and 3742 output units. Each hypothesis in the 100-best list was re-scored using the RNNLMs and LSTMLMs. In addition, the language model scores obtained

TABLE III
Setup for NTT meeting corpus.

	training	dev.	test
# of utterances	20176	4748	4646
# of sessions	40	8	8
vocabulary size	6050	2409	2351
# of words	150215	31074	30371

TABLE IV
Setup for AMI meeting corpus.

	training	dev.	test
recording time (h)	81	9	9
# of utterances	108503	13098	12643
vocabulary size	11883	4146	3913
# of words	802894	94953	89666

TABLE V
RNNLM PARAMETERS THAT WERE USED.

# of hidden units	40, 50, 70, 100, 200, 300
BPTT	1, 2, 4, 8, 10
learning rate	0.1
l^2-regularization parameter	1.0×10^{-5}

TABLE VI
LSTMLM PARAMETERS THAT WERE USED.

# of hidden units	300, 600, 1000
BPTT	5, 10, 15, 20
learning rate	0.1
l^2-regularization parameter	1.0×10^{-5}
dropout rate	0.5

TABLE VII
TEST SET PERPLEXITIES ON NTT MEETING CORPUS. "EOS" MEANS END-OF-SENTENCE AND "SA" MEANS SPEAKER ALTERNATIONS.

model	context cue	feature vector	PPL
RNN			55.3
RNN_CE	EOS		49.5
RNN_E		EOS	50.2
RNN_CE_E	EOS	EOS	50.3
RNN_CEA	EOS & SA		**48.5**
RNN_EA		EOS & SA	48.8

TABLE VIII
TEST SET PERPLEXITIES ON AMI MEETING CORPUS. "EOS" MEANS END-OF-SENTENCES AND "SA" MEANS SPEAKER ALTERNATIONS.

model	context cue	feature vector	PPL
LSTM			82.9
LSTM_CE	EOS		73.4
LSTM_E		EOS	73.6
LSTM_CE_E	EOS	EOS	74.0
LSTM_CEA	EOS & SA		**73.2**
LSTM_EA		EOS & SA	73.5

from the RNNLMs and LSTMLMs were linearly interpolated with the scores obtained from the 3-gram language model, where the interpolation coefficients for the RNNLM/LSTMLM and those for the 3-gram language model were both set to one for the NTT corpus, and set to one and four for the AMI corpus.

These tables summarize the five main results obtained on the RNNLMs for the multiparty conversations:

- The results on the NTT meeting corpus and those on the AMI meeting corpus had similar trends where the information on the end-of-sentences and speaker alternations could be effective in RNNLMs and LSTMLMs for multiparty conversations, irrespective of the spoken languages or data size,
- Exploiting the information on speaker alternations in addition to the generally used end-of-sentences reduced the perplexity and word error rate,
- The context cue and auxiliary feature representations yielded no clear differences in the performance of RNNLMs or LSTMLMs,
- The results obtained from **RNN_CE_E** and **LSTM_CE_E** suggest that simultaneous use of both context cues and feature vectors was redundant and did not help in improving performance, and
- The results in terms of both the word error rate and perplexity indicated that **RNN_CEA** and **LSTM_CEA**, which exploited information on the end-of-sentences and speaker alternations that was represented by context cues, could be the most effective tools in speech recognition and word prediction.

TABLE IX
WORD ERROR RATES (WERS) (%) ON NTT MEETING CORPUS. "EOS" MEANS END-OF-SENTENCES AND "SA" MEANS SPEAKER ALTERNATIONS.

model	context cue	feature vector	WER
w/o re-scoring	-	-	20.8
RNN			19.6
RNN_CE	EOS		19.6
RNN_E		EOS	19.6
RNN_CE_E	EOS	EOS	19.6
RNN_CEA	EOS & SA		**19.5**
RNN_EA		EOS & SA	19.7

TABLE X
WORD ERROR RATES (%) ON AMI MEETING CORPUS. "EOS" MEANS END-OF-SENTENCES AND "SA" MEANS SPEAKER ALTERNATIONS.

model	context cue	feature vector	WER
w/o re-scoring	-	-	24.5
LSTM			23.9
LSTM_CE	EOS		23.8
LSTM_E		EOS	**23.7**
LSTM_CE_E	EOS	EOS	23.8
LSTM_CEA	EOS & SA		**23.7**
LSTM_EA		EOS & SA	23.8

D. Discussion

The same experiment was carried out on transcriptions of spoken lectures, which were different from multiparty conversations to emphasize the importance of handling end-of-sentences in RNNLMs in multiparty conversations. We randomly chose 320 spoken lectures in this case from the corpus of spontaneous Japanese (CSJ) [24], and LSTMLMs that deal with end-of-sentences were evaluated in terms of perplexity. The use of end-of-sentences reduced the perplexity by 6.1% on the CSJ corpus in the best case and reduced the perplexities by 10.5% on the NTT meeting corpus and by 11.5% on the AMI meeting corpus. This suggests that the consideration of end-of-sentences in RNNLMs is important, especially in multiparty conversations.

V. CONCLUSION

The use of information on end-of-sentences and speaker alternations in RNNLMs was discussed to develop effective language models for multiparty conversations. Two attempts were specifically made to represent such auxiliary information: the first representation was the use of context cues as other input tokens and the second involved feature vectors taken as inputs to neural networks at individual time steps. Experimental comparisons conducted on Japanese and English meeting data demonstrated that exploiting information on speaker alternations yielded improvements over when only generally-used end-of-sentences were used, but information on end-of-sentences were dominant in performance. Two representations for auxiliary information in this case did not create large differences in performance. In addition, it was important to handle information on end-of-sentences in language modeling, especially that for multiparty conversations.

REFERENCES

[1] G. Hinton and *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process.*, vol. 29, no. 6, pp. 82–97, 2012.

[2] M. Nelson and *et al.*, "Meetings about meetings: research at ICSI on speech in multiparty conversations," in *Proc. ICASSP*, April 2003, pp. 740–743.

[3] T. Hori and *et al.*, "Real-time meeting recognition and understanding using distant microphones and omni-directional camera," in *Proc. SLT*, Dec. 2010, pp. 424–429.

[4] T. Hain and *et al.*, "Transcribing meetings with the AMIDA systems," *IEEE Trans. Audio, Speech & Lang. Process.*, no. 2, pp. 486–498, 2012.

[5] W. Xu and A. Rudnicky, "Language modeling for dialog system," in *Proc. ICSLP*, Oct. 2000, pp. 118–121.

[6] K. Yoshino, S. Mori, and T. Kawahara, "Language modeling for spoken dialogue system based on sentence transformation and filtering using predicate-argument structures," in *Proc. APSIPA*, Dec. 2012, pp. 1–4.

[7] A. Sapru and H. Bourlard, "Detecting speaker roles and topic changes in multiparty conversations using latent topic models," in *Proc. INTERSPEECH*, Sept. 2014, pp. 2882–2886.

[8] F. Valente and A. Vinciarelli, "Language-independent socio-emotional role recognition in the AMI meetings corpus," in *Proc. INTERSPEECH*, Aug. 2011, pp. 3077–3080.

[9] A. Sapru and F. Valente, "Automatic speaker role labeling in AMI meetings: recognition of formal and social roles," in *Proc. ICASSP*, March 2012, pp. 5057–5060.

[10] D. Huggins-Daines and A. I. Rudnicky, "Implicitly supervised language model adaptation for meeting transcription," in *Proc. NAACL HLT*, April 2007, pp. 73–76.

[11] Y. Tam and P. Vozila, "Unsupervised latent speaker language modeling," in *Proc. INTERSPEECH*, Aug. 2011, pp. 1477–1480.

[12] R. Masumura, T. Oba, H. Masataki, O. Yoshioka, and S. Takahashi, "Role play dialogue topic model for language model adaptation in multi-party conversation speech recognition," in *Proc. ICASSP*, May 2014, pp. 4873–4877.

[13] N. Murai and T. Kobayashi, "Dictation of multiparty conversation using statistical turn taking model and speaker model," in *Proc. ICASSP*, June 2000, pp. 1575–1578.

[14] R. Sarikaya, Y. Gao, H. Erdogan, and M. Picheny, "Turn-based language modeling for spoken dialog systems," in *Proc. ICASSP*, May 2002, pp. 781–784.

[15] G. Ji and J. Bilmes, "Multi-speaker language modeling," in *Proc. NAACL HLT*, May 2004, pp. 133–136.

[16] G. Ji and J. A. Bilmes, "Jointly recognizing multi-speaker conversations," in *Proc. ICASSP*, March 2010, pp. 5110–5113.

[17] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. INTERSPEECH*, Sept. 2010, pp. 1045–1048.

[18] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling," in *INTERSPEECH*, Sept. 2012, pp. 194–197.

[19] P. J. Werbos, "Backpropagation through time: what does it do and how to do it," in *Proceedings of IEEE*, vol. 78, no. 10, 1990, pp. 1550–1560.

[20] J. Carletta and *et al.*, "The AMI meeting corpus: A pre-announcement," in *Proc. MIML*, 2006, pp. 28–39.

[21] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *JMLR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Trans. Audio, Speech & Lang. Process.*, vol. 15, no. 4, pp. 1352–1365, 2007.

[23] D. Povey and *et al.*, "The kaldi speech recognition toolkit," in *Proc. ASRU*, Dec. 2011.

[24] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of japanese," in *Proc. LREC*, 2000.