Towards Neural Speaker Modeling in Multi-Party Conversation: The Task, Dataset, and Models

Zhao Meng,[1,2] **Lili Mou,**[1,3] **Zhi Jin**[1,*]
[1]Key Laboratory of High Confidence Software Technologies, MoE; Software Institute, Peking University
[2]Department of Computer Science, ETH Zurich
[3]David R. Cheriton School of Computer Science, University of Waterloo
zhmeng@student.ethz.ch, doublepower.mou@gmail.com, zhijin@sei.pku.edu.cn

Abstract

Neural network-based dialog systems are attracting increasing attention in both academia and industry. Recently, researchers have begun to realize the importance of speaker modeling in neural dialog systems, but there lacks established tasks and datasets. In this paper, we propose *speaker classification* as a surrogate task for general speaker modeling, and collect massive data to facilitate research in this direction. We further investigate temporal-based and content-based models of speakers, and propose several hybrids of them. Experiments show that speaker classification is feasible, and that hybrid models outperform each single component.

Keywords: Speaker Classification, Speaker Modeling, Multi-Party Conversation

1. Introduction

Human-computer conversation has long attracted attention in both academia and industry. Researchers have developed a variety of approaches, ranging from rule-based systems for task-oriented dialog (Ferguson et al., 1996; Graesser et al., 2005) to data-driven models for open-domain conversation (Ritter et al., 2011).

A simple setting in the research of dialog systems is context-free, i.e., only a single utterance is considered during reply generation (Shang et al., 2015). Other studies leverage context information by concatenating several utterances (Sordoni et al., 2015) or building hierarchical models (Yao et al., 2015; Serban et al., 2016). The above approaches do not distinguish different speakers, and thus speaker information would be lost during conversation modeling.

Speaker modeling is in fact important to dialog systems, and has been studied in traditional dialog research. However, existing methods are usually based on hand-crafted statistics and *ad hoc* to a certain application (Lin and Walker, 2011). Another research direction is speaker modeling in a multi-modal setting, e.g., acoustic and visual (Uthus and Aha, 2013), which is beyond the focus of this paper.

Recently, neural networks have become a prevailing technique in both task-oriented and open-domain dialog systems. After single-turn and multi-turn dialog studies, a few researchers have realized the role of speakers in neural conversational models. Li et al. (2016) show that, with speaker identity information, a sequence-to-sequence neural dialog system tends to generate more coherent replies. In their approach, a speaker is modeled by a learned vector (also known as an *embedding*). Such method, unfortunately, requires massive conversational data for a particular speaker to train his/her embedding, and thus does not generalize to rare or unseen speakers.

Ouchi and Tsuboi (2016) formalize a new task of addressee selection on online forums: by leveraging either the temporal or utterance information, they predict whom a post is talking to. While tempting for benchmarking speaker modeling, the task requires explicit speaker ID mentions, which occurs occasionally, and thus is restricted.

In this paper, we propose a *speaker classification* task that predicts the speaker of an utterance. It serves as a surrogate task for general speaker modeling, similar to *next utterance classification* (Lowe et al., 2015, NUC) being a surrogate task for dialog generation. The speaker classification task could also be useful in applications like *speech diarization*,[1] where text understanding can improve speaker segmentation, identification, etc. in speech processing (Li et al., 2009; Meng et al., 2017).

We further propose a neural model that combines temporal and content information with interpolating or gating mechanisms. The observation is that, what a speaker has said (called `content`) provides non-trivial background information of the speaker. Meanwhile, the relative order of a speaker (e.g., the i-th latest speaker) is a strong bias: nearer speakers are more likely to speak again; we call it `temporal` information. We investigate different strategies for combination, ranging from linear interpolation to complicated gating mechanisms inspired by Differentiable Neural Computers (Graves and others, 2016, DNC).

To evaluate our models, we constructed a massive corpus using transcripts of TV talk shows from the Cable News Network website. Experiments show that combining content and temporal information significantly outperforms either of them, and that simple interpolation is surprisingly more efficient and effective than gating mechanisms.

Datasets and code are available on our project website.[2]

2. Task Formulation and Data Collection

We formulate speaker classification as follows.

Assume that we have segmented a multi-party conversation into several parts by speakers; each segment com-

[*]Corresponding author.

[1]Speech diarization aims at answering "who spoke when" (Anguera et al., 2012).

[2]`https://sites.google.com/site/neuralspeaker/`

Data partition	# of samples
Train	174,487
Validation	21,071
Test	20,501

Table 1: Dataset statistics.

prises one or a few consecutive sentences $u_1, u_2, \cdots, u_N$, uttered by a particular speaker. A candidate set of speakers $\mathcal{S} = \{s_1, s_2, \cdots, s_k\}$ is also given. In our experiments, we assume $u_1, u_2, \cdots, u_N$'s speaker s_i is in $\mathcal{S}$. The task of speaker classification is to identify the speaker s_i of $u_1, \cdots, u_N$.

Following the spirit of distributed semantics (e.g., word embeddings), we represent the current utterance(s) as a real-valued vector $\boldsymbol{u}$ with recurrent neural networks. Speakers are also represented as vectors $\boldsymbol{s}_i, \cdots, \boldsymbol{s}_k$. The speaker classification is accomplished by a softmax-like function

$$\widetilde{p}_i = \exp\left\{\boldsymbol{s}_i^\top \boldsymbol{u}\right\} \tag{1}$$

$$p(s_i) = \frac{\widetilde{p}_i}{\sum_j \widetilde{p}_j} \tag{2}$$

Because the number of candidate speakers may vary, the "weights" of softmax are not a fixed-size matrix, but the distributed representations of candidate speakers, $\boldsymbol{s}_1, \cdots, \boldsymbol{s}_k$. In Section 3., we investigate several approaches of modeling $\boldsymbol{s}_i$ based on what a speaker says or the relative order of a speaker in the dialog; we also propose to combine them by interpolating or gating mechanisms.

To facilitate the speaker classification task, we crawled transcripts of more than 8,000 episodes of TV talk shows.[3] We assumed that the current speaker is within the nearest k speakers. ($k = 5$, but at the beginning, k may be less than 5.) Since too few utterances do not provide much information, we required each speaker having at least 3 previous utterances, but kept at most 5. Samples failing to meet the above requirements were filtered out during data preprocessing.

We split train/val/test sets according to TV show episodes instead of sentences; therefore no utterance overlaps between training and testing. Table 1 shows the statistics of our dataset partition.

3. Methodology

We use a hierarchical recurrent neural network (Serban et al., 2016) to model the current utterances $u_1, \cdots, u_N$ (Figure 1a). In other words, a recurrent neural network (RNN) captures the meaning of a sentence; another LSTM-RNN aggregates the sentence information into a fixed-size vector. For simplicity, we use RNN's last state as the current utterances' representation ($\boldsymbol{u}$ in Equation 2).

In the rest of this section, we investigate content-based and temporal-based prediction in Subsections 3.1. and 3.2.; the spirit is similar to "dynamic" and "static" models, respectively, in Ouchi and Tsuboi (2016). We combine content-based and temporal-based prediction using gating mechanisms in Subsection 3.3..

3.1. Prediction with Content Information

In this method, we model a speaker by what he or she has said, i.e., content.

Figure 1b illustrates the content-based model: a hierarchical RNN (which is the same as Figure 1a) yields a vector $\boldsymbol{s}_i$ for each speaker, based on his or her nearest several utterances. The speaker vector $\boldsymbol{s}_i$ is multiplied by current utterances' vector $\boldsymbol{u}$ for softmax-like prediction (Equation 2). We pick the candidate speaker that has the highest probability.

It is natural to model a speaker by his/her utterances, which provide illuminating information of the speaker's background, stance, etc. As will be shown in Section 4., content-based prediction achieves significantly better performance than random guess. This also verifies that *speaker classification* is feasible, being a meaningful surrogate task for speaker modeling.

3.2. Prediction with Temporal Information

In temporal-based approach, we sort all speakers in a descending order according to the last time he or she speaks, and assign a vector (embedding) for each index in the list, following the "static model" in Ouchi and Tsuboi (2016). Each speaker vector is randomly initialized and optimized as parameters during training. The predicted probability of a speaker is also computed by Equation 2.

The temporal vector is also known as a *position embedding* in other NLP literature (Nguyen and Grishman, 2015). Our experiments show that temporal information provides strong bias: nearer speakers tend to speak more; hence, it is also useful for speaker modeling.

3.3. Combining Content and Temporal Information

As both content and temporal provide important evidence for speaker classification, we propose to combine them by interpolating or gating mechanisms (illustrated in Figure 1d). In particular, we have

$$\boldsymbol{p}^{(\text{hybrid})} = (1 - g) \cdot \boldsymbol{p}^{(\text{temporal})} + g \cdot \boldsymbol{p}^{(\text{content})} \tag{3}$$

Here, g is known as a *gate*, balancing these two aspects. We investigate three strategies to compute the gate.

1. **Interpolating after training.** The simplest approach, perhaps, is to train two predictors separately, and interpolate after training by validating the hyperparameter g.

2. **Interpolating while training.** We could also train the hybrid model as a whole with cross-entropy loss directly applied to Equation 3.

3. **Self-adaptive gating.** Inspired by hybrid content- and location-based addressing in Differentiable Neural Computers (Graves and others, 2016, DNCs), we design a learnable gate in hopes of dynamically balancing temporal and content information. Different

[3] https://transcripts.cnn.com

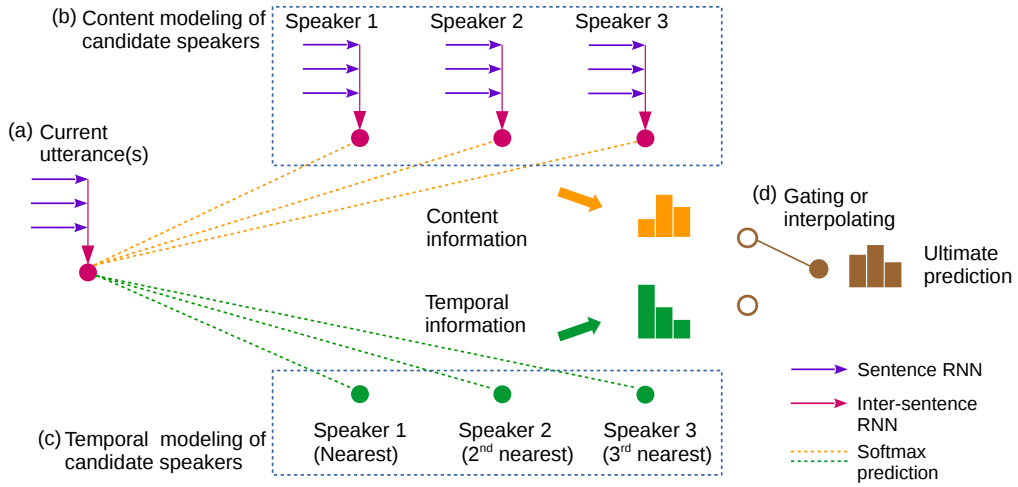

Figure 1: Hybrid content- and temporal-based speaker classification with a gating mechanism.

Model	Macro F_1	Weighted F_1	Micro F_1	Acc.	MRR.
Random guess	19.93	34.19	27.53	27.53	N/A
Majority guess	21.26	62.96	74.01	74.01	N/A
Hybrid random/majority guess	25.26	61.99	69.29	69.29	N/A
Temporal information	26.07	63.60	73.99	73.99	84.85
Content information	42.61	65.04	61.82	58.58	74.86
+ static attention	42.50	65.28	61.79	58.99	74.89
+ sentence-by-sentence attention	42.56	65.96	62.86	59.81	75.58
Hybrid Interpolating after training	**44.25**	**71.35**	**76.10**	**75.84**	**85.73**
Hybrid Interpolating while training	41.30	70.10	75.57	75.31	85.20
Hybrid Self-adaptive gating	39.45	69.55	74.11	74.09	84.85

Table 2: Model performance (in percentage).

from DNCs, however, the gate here is not based on input (i.e., $\boldsymbol{u}$ in our scenario), but the result of content prediction $\boldsymbol{p}^{(\text{content})}$. Formally

$$g = \text{sigmoid}\left(w \cdot \text{std}[\,p^{(\text{content})}\,] + b\right) \qquad (4)$$

where we compute the standard deviation (std) of p. w and b are parameters to scale $\text{std}[\,p^{(\text{content})}\,]$ to a sensitive region of the sigmoid function.

4. Experimental Results

Setup. All our neural layers including word embeddings were set to 100-dimensional. We tried larger dimensions, resulting in slight but insignificant improvement. We did not use pretrained word embeddings but instead randomly initialized them because our dataset is large. We used the Adam optimizer (Kingma and Ba, 2015) mostly with default hyperparameters. We set the batch size to 10 due to GPU memory constraints. Dropout rate and early stop were also applied by validation. Notice that validation was accomplished by each metric itself because different metrics emphasize different aspects of model performance.

Performance. Table 2 compares the performance of different models. Majority-class guess results in high accuracy, showing that the dataset is screwed. Hence, we choose macro F_1 as our major metric, which addresses minority classes more than other metrics. We nevertheless present other metrics including accuracy, mean reciprocal ranking (MRR), and micro/weighted F_1 as additional evidence.

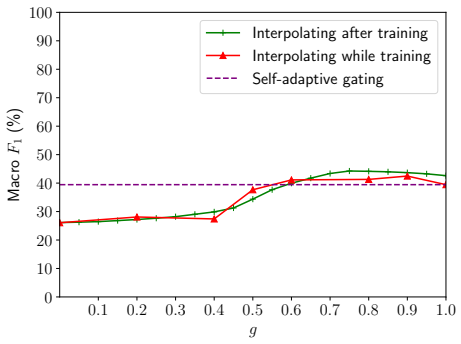

Figure 2: Performance vs. the hyperparameter g (solid lines). $g = 0$: temporal only; $g = 1$: content only. The self-adaptive gating mechanism is also plotted for comparison (which is not associated with a particular value of g).

As shown, the content-based model achieves higher performance in macro F_1 than majority guess, showing the effectiveness of content information. Following Rocktäschel et al. (2016), we adopt a static or sentence-by-sentence attention mechanism. The LSTM-RNN attends to speaker s_i to obtain speaker vector $\boldsymbol{s}_i$ while it is encoding current utterances. However, such attention mechanisms bring little improvements (if any). Hence, we do not use attention in our hybrid models for simplicity.

All hybrid models achieve higher performance compared with either content- or temporal-based prediction in terms

of most measures, which implies content and temporal information sources capture different aspects of speakers.

Among different strategies of hybrid models, the simple approach "interpolating after training" surprisingly outperforms the other two. A plausible explanation is that training a hybrid model as a whole leads to optimization difficulty in our scenario; that simply interpolating well-trained models is efficient yet effective. However, the hyperparameter g is sensitive and only yields high performance in the range $(0.6, 0.9)$. Thus, the learnable gating mechanism could also be useful in some scenarios, as it is self-adaptive.

5. Conclusion and Future Work

In this paper, we addressed the problem of neural speaker modeling in multi-party conversation. We proposed *speaker classification* as a surrogate task and collected massive TV talk shows as our corpus. We investigated content-based and temporal-based models, as well as their hybrids. Experimental results show that speaker classification is feasible, being a meaningful task for speaker modeling; that interpolation between content- and temporal-based prediction yields the highest performance.

In the future, we would like to design more dedicated gating mechanisms to improve the performance; we would also like to explore other aspects of speaker modeling, e.g., incorporating dialog context before current utterances. The collected dataset is also potentially useful in other applications.

6. Bibliographical References

Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., and Vinyals, O. (2012). Speaker diarization: A review of recent research. *IEEE Transactions on Audio Speech and Language Processing*, 20(2):356–370.

Ferguson, G., Allen, J. F., and Miller, B. (1996). Trains-95: Towards a mixed-initiative planning assistant. In *Proceedings of Artificial Intelligence Planning Systems Conference*, pages 70–77.

Graesser, A. C., Chipman, P., Haynes, B. C., and Olney, A. (2005). AutoTutor: an intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618.

Graves, A. et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the International Conference for Learning Representations*.

Li, R., Schultz, T., and Jin, Q. (2009). Improving speaker segmentation via speaker identification and text segmentation. In *INTERSPEECH*, pages 904–907.

Li, J., Galley, M., Brockett, C., Spithourakis, G., Gao, J., and Dolan, B. (2016). A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 994–1003.

Lin, G. I. and Walker, M. A. (2011). All the world's a stage: Learning character models from film. In *Proceedings of the Seventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, pages 46–52.

Lowe, R., Pow, N., Serban, I., and Pineau, J. (2015). The Ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Meng, Z., Mou, L., and Jin, Z. (2017). Hierarchical RNN with static sentence-level attention for text-based speaker change detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2203–2206. ACM.

Nguyen, T. H. and Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Ouchi, H. and Tsuboi, Y. (2016). Addressee and response selection for multi-party conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2133–2143.

Ritter, A., Cherry, C., and Dolan, W. B. (2011). Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 583–593.

Rocktäschel, T., Grefenstette, E., Hermann, K. M., Kočiskỳ, T., and Blunsom, P. (2016). Reasoning about entailment with neural attention. In *Proceedings of the International Conference on Learning Representations*.

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2016). Building end- to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 3776–3783.

Shang, L., Lu, Z., and Li, H. (2015). Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586, July.

Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.-Y., Gao, J., and Dolan, B. (2015). A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205.

Uthus, D. C. and Aha, D. W. (2013). Multiparticipant chat analysis: A survey. *Artificial Intelligence*, 199:106–121.

Yao, K., Zweig, G., and Peng, B. (2015). Attention with intention for a neural network conversation model. *arXiv preprint arXiv:1510.08565*.