How to Make Context More Useful?
An Empirical Study on Context-Aware Neural Conversational Models

Zhiliang Tian,[1] Rui Yan,[2*] Lili Mou,[3] Yiping Song,[4] Yansong Feng,[2] Dongyan Zhao[2]
[1]Baidu Inc., China tianzhiliang@baidu.com
[2]Institute of Computer Science and Technology, Peking University, China
[3]Key Laboratory of High Confidence Software Technologies, MoE, China
Institute of Software, Peking University, China
[4]Institute of Network Computing and Information Systems, Peking University, China
{ruiyan,songyiping,yansong,zhaody}@pku.edu.cn doublepower.mou@gmail.com

Abstract

Generative conversational systems are attracting increasing attention in natural language processing (NLP). Recently, researchers have noticed the importance of context information in dialog processing, and built various models to utilize context. However, there is no systematic comparison to analyze how to use context effectively. In this paper, we conduct an empirical study to compare various models and investigate the effect of context information in dialog systems. We also propose a variant that explicitly weights context vectors by context-query relevance, outperforming the other baselines.

1 Introduction

Recently, human-computer conversation is attracting increasing attention due to its promising potentials and alluring commercial values. Researchers have proposed both retrieval methods (Ji et al., 2014; Yan et al., 2016) and generative methods (Ritter et al., 2011; Shang et al., 2015) for automatic conversational systems. With the success of deep learning techniques, neural networks have demonstrated powerful capability of learning human dialog patterns; given a user-issued utterance as an input query q, the network can generate a reply r, which is usually accomplished in a sequence-to-sequence (Seq2Seq) manner (Shang et al., 2015).

In the literature, there are two typical research setups for dialog systems: single-turn and multi-turn. Single-turn conversation is, perhaps, the simplest setting where the model only takes q into consideration when generating r (Shang et al., 2015; Mou et al., 2016). However, most real-world dialogs comprise multiple turns. Previous utterances (referred to as *context* in this paper) could also provide useful information about the dialog status and are the key to coherent multi-turn conversation.

Existing studies have realized the importance of context, and proposed several context-aware conversational systems. For example, Yan et al. (2016) directly concatenate context utterances and the current query; others use hierarchical models, first capturing the meaning of individual utterances and then integrating them as discourses (Serban et al., 2016). There could be several ways of combining context and the current query, e.g., pooling or concatenation (Sordoni et al., 2015). Unfortunately, previous literature lacks a systematic comparison of the above methods.

In this paper, we conduct an empirical study on context modeling in Seq2Seq-like conversational systems. We emphasize the following research questions:

- **RQ1**. *How can we make better use of context information?* Our study shows that hierarchical models are generally better than non-hierarchical ones. We also propose a variant of context integration that explicitly weights a context vector by its relevance measure, outperforming simple vector pooling or concatenation.
- **RQ2**. *What is the effect of context on neural dialog systems?* We find context information is useful to neural conversational models. It yields longer, more informative and diversified replies.

To sum up, the contributions of this paper are two-fold: (1) We conduct a systematic study on context modeling in neural conversational models. (2) We further propose an explicitly con-

[*] Corresponding author.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 231–236
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-2036

text weighting approach, outperforming the other baselines.

2 Models

2.1 Non-Hierarchical Model

To model a few utterances before the current query, several studies directly concatenate these sentences together and use a single model to capture the meaning of context and the query (Yan et al., 2016; Sordoni et al., 2015). They are referred to as *non-hierarchical models* in our paper. Such method is also used in other NLP tasks, e.g., document-level sentiment analysis (Xu et al., 2016) and machine comprehension (Wang and Jiang, 2017).

Following the classic encode-decoder framework, we use a `Seq2Seq` network, which transforms the query and context into a fixed-length vector $\boldsymbol{v}_{\text{enc}}$ by a recurrent neural network (RNN) during encoding; then, in the decoding phase, it generates a reply r with another RNN in a word-by-word fashion. (See Figure 1a.)

In our study, we adopt RNNs with gated recurrent units (Cho et al., 2014, GRUs), which alleviates the long propagation problem of vanilla RNNs. When decoding, we apply beam search with a size of 5.

2.2 Hierarchical Model

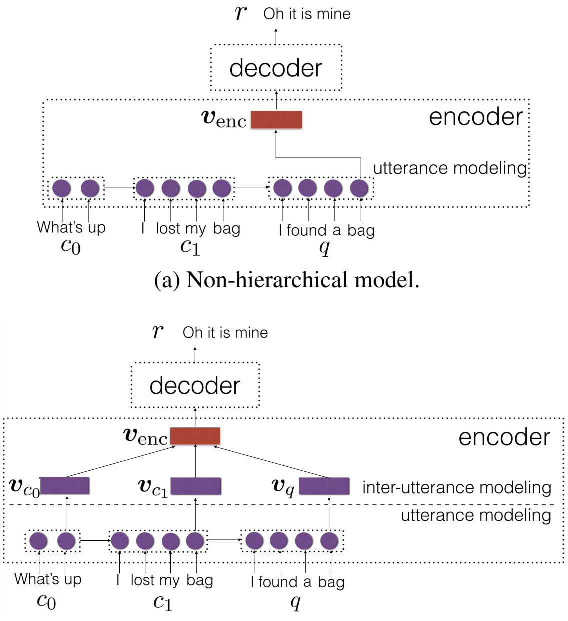
A more complicated approach to context modeling is to build hierarchical models with a two-step strategy: an utterance-level model captures the meaning of each individual sentences, and then an inter-utterance model integrates context and query information (Figure 1b).

Researchers have tried different ways of combining information during inter-utterance modeling; this paper evaluates several prevailing methods.

Sum pooling. Sum pooling (denoted as `Sum`) integrates information over a candidate set by summing the values in each dimension (Figure 2a). Given context vectors $\boldsymbol{v}_{c_1}, \cdots, \boldsymbol{v}_{c_n}$ and the query vector $\boldsymbol{v}_q$, the encoded vector $\boldsymbol{v}_{\text{enc}}$ is

$$\boldsymbol{v}_{\text{enc}} = \sum_{i=1}^{n} \boldsymbol{v}_{c_i} + \boldsymbol{v}_q \tag{1}$$

Sum pooling is used in Sordoni et al. (2015), where bag-of-words (BoW) features of context and the query are simply added. In our experiments, sum pooling operates on the features extracted by sentence-level RNNs of context and query utterances, as modern neural networks preserve more information than BoW features.



(a) Non-hierarchical model.

(b) Hierarchical model.

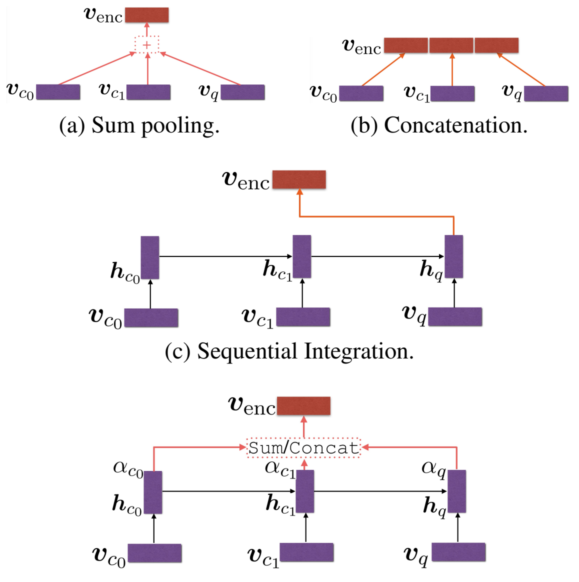
Figure 1: `Seq2Seq`-like neural networks generate a reply r based on context $\mathcal{C} = \{c_1, \ldots, c_n\}$ and the current query q with (a) non-hierarchical or (b) hierarchical models.

Concatenation. Concatenation (`Concat`) is yet another method used in Sordoni et al. (2015). This strategy concatenates every utterance-level vectors $\boldsymbol{v}_{c_i}$ and $\boldsymbol{v}_q$ as a long vector, i.e., $\boldsymbol{v}_{\text{enc}} = [\boldsymbol{v}_{c_0}; \ldots; \boldsymbol{v}_{c_n}; \boldsymbol{v}_q]$. (See Figure 2b.)

Compared with sum pooling, vector concatenation can distinguish different roles of the context and query, as this operation keeps input separately. One potential shortcoming, however, is that concatenation only works with fixed-length context.

Sequential integration. Yao et al. (2015) and Serban et al. (2015) propose hierarchical dialog systems, where an inter-utterance RNN is built upon utterance-level RNNs' features (last hidden state). Training is accomplished by end-to-end gradient propagation, and the process is illustrated in Figure 2c.

Using an RNN to integrate context and query vectors in a sequential manner enables complicated information interaction. Based on the RNN's hidden states, `Sum` and `Concat` could also be applied to obtain the encoded vector $\boldsymbol{v}_{\text{enc}}$.



(a) Sum pooling.

(b) Concatenation.

(c) Sequential Integration.

(d) Weighted Sequential Integration.

Figure 2: The inter-utterance modeling in hierarchical models. v_{c_i} and v_q are the utterance-level vectors, h_{c_i} and h_q are the utterance-level hidden states, α_{c_i} and α_q are the explicitly weights and v_{enc} is the output of the encoder.

However, we found their performance is worse than only using the last hidden state (denoted as `Seq`). One plausible reason might be that the inter-sentence RNN is not long and that RNN can preserve these information well. Therefore, this variant is adopted in our experiments, as shown in Figure 2c.

2.3 Explicitly Weighting by Context-Query Relevance

In conversation, context utterances may vary in content and semantics: context utterances that are relevant to the query may be useful, while irrelevant ones may bring more about noise. Following this intuition, we propose a variant that explicitly weights the context vector by an attention score of context-query relevance.

First, we compute the similarity between the context and query by the cosine measure

$$s_{c_i} = \text{sim}(c_i, q) = \frac{\boldsymbol{e}_{c_i} \cdot \boldsymbol{e}_q}{\|\boldsymbol{e}_{c_i}\| \cdot \|\boldsymbol{e}_q\|} \tag{2}$$

where

$$\boldsymbol{e}_{c_i} = \sum_{w \in c_i} \boldsymbol{e}_w \quad \text{and} \quad \boldsymbol{e}_q = \sum_{w' \in q} \boldsymbol{e}_{w'} \tag{3}$$

that is, the sentence embedding is the sum of word embeddings.

Following the spirit of attention mechanisms (Bahdanau et al., 2014), we would like to normalize these similarities by a softmax function and obtain attention probabilities:

$$\alpha_{c_i} = \frac{\exp(s_{c_i})}{\sum_{j=0}^{n} \exp(s_{c_j}) + \exp(s_q)} \tag{4}$$

$$\alpha_q = \frac{\exp(s_q)}{\sum_{j=0}^{n} \exp(s_{c_j}) + \exp(s_q)} \tag{5}$$

where s_q is computed in the same manner as s_{c_i} and is always 1, which is the cosine of two same vectors. The intuition is that, if the context is less relevant, we should mainly focus on the query itself, but if the context is relevant, we should focus more evenly across context and the query.

In other words, our explicitly weighting approach could be viewed as *heuristic attention*. Akin to Subsection 2.2, we aggregate the weighted context and query vectors by pooling and concatenation, resulting in the following two variants.

- `WSeq (sum)`, where weighted vectors are summed together

$$\boldsymbol{v}_{\text{enc}} = \sum_{i=0}^{n} \alpha_{c_i} \boldsymbol{h}_{c_i} + \alpha_q \boldsymbol{h}_q \tag{6}$$

- `WSeq (concat)`, where weighted vectors are concatenated

$$\boldsymbol{v}_{\text{enc}} = [\alpha_{c_0} \boldsymbol{h}_{c_0}; \ldots; \alpha_{c_n} \boldsymbol{h}_{c_n}; \alpha_q \boldsymbol{h}_q] \tag{7}$$

Notice that the explicitly weighting approach can also be applied to sentence embeddings (without inter-sentence RNN). We denote the variants by `WSum` and `WConcat`, respectively; details are not repeated. They are included for comparison in Section 3.2.

3 Experiments

3.1 Setup

We conducted all experiments on a Chinese dataset crawled from an online free chatting platform, Baidu Tieba.[1] To facilitate the research of context's effect, we established a multi-turn conversational corpus following Sordoni et al. (2015) and Serban et al. (2015). A data sample contains three utterances, being a triple ⟨*last_context, query, reply*⟩. In total, we had

[1] https://tieba.baidu.com

Method		BLEU-1	BLEU-2	BLEU-3	BLEU-4
Context-Insensitive		4.611	1.488	0.657	0.311
Non-Hierarchical		4.805	1.507	0.678	0.343
Hierarchical	`Sum`	4.440	1.367	0.505	0.042
	`WSum`	5.055	1.667	0.741	0.378
	`Concat`	5.107	1.688	0.747	0.420
	`WConcat`	5.181	1.763	0.745	0.342
	`Seq`	**5.355**	1.771	0.916	0.387
	`WSeq (sum)`	5.134	1.586	0.7429	0.4359
	`WSeq (concat)`	5.322	**1.883**	**0.9966**	**0.6897**

Table 1: Performance of different models.

500,000 samples for training, 2000 for validation, and 4000 for testing. The hyperparameters of neural networks were mainly derived from Shang et al. (2015) and Song et al. (2016): embeddings 620d and hidden states 1000d; we used AdaDelta for optimization.

3.2 Results and Analysis

We evaluated model performance by BLEU scores. As this paper compares various models, it is unaffordable for us to hire workers to manually annotate their satisfaction. BLEU scores, albeit imperfect for open-domain dialog systems, exhibits more or less correlation with human satisfaction (Liu et al., 2016; Tao et al., 2017). We present in Table 1 the overall performance of the models introduced in Section 2, and answer our research questions as follows.

RQ1: *How can we make better use of context information?*

We first observe that context-aware methods generally outperform the context-insensitive one. This implies context is indeed useful in open-domain, chit-chat-style dialog systems. The results are consistent with previous studies (Sordoni et al., 2015; Serban et al., 2015).

Among context-aware neural conversational models, we have the following findings.

- Hierarchical structures outperform the non-hierarchical one.

  Comparing the non-hierarchical and hierarchical structures, we find it obvious that (most) hierarchical models outperform the non-hierarchical one by a large margin. The results show that, dialog systems are different from other NLP applications, e.g., comprehension (Wang and Jiang, 2017), where non-hierarchical recurrent neural networks are adopted to better integrate information across different sentences. A plausible explanation, as indicated by Meng et al. (2017), is that conversational sentences are not necessarily uttered by a same speaker, and literature shows consistent evidence of the effectiveness of hierarchical RNNs in dialog systems.

- Keeping the roles of different utterances separately is important.

  As mentioned in Section 2, the concatenation operation (`Concat`) distinguishes the roles of different utterances, while sum pooling `Sum` aggregates information in a homogeneous way. We see that the former outperforms the latter in both sentence-embedding and inter-sentence RNN levels, showing that sum pooling is not suitable for treating dialog context. Our conjecture is that sum pooling buries illuminating query information under less important context. Hence, keeping them separately will generally help.

- The context-query relevance score benefits conversational systems.

  Our explicitly weighting approach computes an attention probability by context-query relevance. In all variants (`Sum`, `Concat`, and `Seq`), explicitly weighting improves the performance by a large margin (except BLEU-1 for `Seq`). The results indicate that context-query relevance is useful, as it emphasizes

Method	Length	Entropy	Diversity
Context-Insensitive	4.008	7.648	0.917
Context-Aware	4.204	7.863	0.927
Ground Truth	9.735	9.277	0.949

Table 2: The length, entropy, and diversity of the replies on the context-insensitive and context-aware (`WSeq, concat`) methods.

relevant context utterances as well as weakens irrelevant contexts.

RQ2: *What is the effect of context on neural dialog systems?*

We are now curious about how context information affects neural conversational systems. In Table 2, we present three auxiliary metrics, i.e., sentence length, entropy, and diversity. The former two are used in Serban et al. (2016) and Mou et al. (2016), whereas the latter one is used in Zhang and Hurley (2008).

As shown, content-aware conversational models tend to generate longer, more meaningful and diverse replies compared with content-insensitive models, given that they also improve BLEU scores.[2]

This shows an interesting phenomenon of neural sequence generation: an encoder-decoder framework needs sufficient source information for meaningful generation of the target; it simply does not fall into meaningful content from less meaningful input. A similar phenomenon is also reported in our previous work (Mou et al., 2016); we show that, a same network will generate more meaningful sentences if it starts from a given (meaningful) keyword. These results also partially explain why a `seq2seq` neural network tends to generate short and universally relevant replies in open-domain conversation, despite its success in machine translation, abstractive summarization, etc.

4 Conclusion

In this work, we analyzed the effect of context in generative conversational models. We conducted a systematic comparison among existing methods and our newly proposed variant that explicitly weights context vectors by context-query relevance.

We show that hierarchical RNNs generally outperform non-hierarchical ones, and that explicitly weighting context information can emphasize the relevant context utterances and weaken less relevant ones.

Our experiments also reveal an interesting phenomenon: with context information, neural networks tend to generate longer, more meaningful and diverse replies, which sheds light on neural sequence generation.

Acknowledgments

We thank the reviewers for insightful comments. This research is partially supported by 863 Grant No. 2015AA015403 and NSFC Grant No. 61672058.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* .

Zongcheng Ji, Zhengdong Lu, and Hang Li. 2014. An information retrieval approach to short text conversation. *arXiv preprint arXiv:1408.6988* .

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 2122–2132. https://doi.org/10.18653/v1/D16-1230.

Zhao Meng, Lili Mou, and Zhi Jin. 2017. Hierarchical RNN with static sentence-level attention for text-based speaker change detection. *arXiv preprint arXiv:1703.07713* .

Lili Mou, Yiping Song, Rui Yan, Ge Li, Lu Zhang, and Zhi Jin. 2016. Sequence to backward and forward sequences: A content-introducing approach to generative short-text conversation. In *Proceedings of the 26th International Conference on Computational Linguistics*. pages 3349–3358. http://aclweb.org/anthology/C16-1316.

[2] This condition is important when we draw conclusions. The length, entropy and diversity metrics do not make sense by themselves alone, because a system can achieve very high scores by repetitively generating random words.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pages 583–593. http://aclweb.org/anthology/D11-1054.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2015. 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.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. A hierarchical latent variable encoder-decoder model for generating dialogues. *arXiv preprint arXiv:1605.06069* .

Lifeng Shang, Zhengdong Lu, and Hang Li. 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. https://doi.org/10.3115/v1/P15-1152.

Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval-and generation-based dialog systems. *arXiv preprint arXiv:1610.07149* .

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 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. https://doi.org/10.3115/v1/N15-1020.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2017. RUBER: An unsupervised method for automatic evaluation of open-domain dialog systems. *arXiv preprint arXiv:1701.03079* .

Shuohang Wang and Jing Jiang. 2017. Machine comprehension using match-LSTM and answer pointer. In *Proceedings of the International Conference on Learning Representations*.

Jiacheng Xu, Danlu Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Cached long short-term memory neural networks for document-level sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. pages 1660–1669. https://doi.org/10.18653/v1/D16-1172.

Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. pages 55–64.

Kaisheng Yao, Geoffrey Zweig, and Baolin Peng. 2015. Attention with intention for a neural network conversation model. In *NIPS Workshop*.

Mi Zhang and Neil Hurley. 2008. Avoiding monotony: Improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems*. pages 123–130.