

How to Make Context More Useful?

An Empirical Study on Context-Aware Neural Conversational Models

# Abstract

最近许多研究者都注意到了上下文在对话系统中的重要性，也做了很多的研究。

但是没有系统的比较来分析怎样才能有效地利用上下文。

我们做了详细的研究来比较不同的模型，研究上下文在对话系统中的作用。

同时，我们提出了度量上下文与查询之间相关性的方法，比其他方法性能都出色。

对话系统有两种典型的研究设置：单轮和多轮。

单轮就是只输入查询 $q$ ，输出答案 $r$ 。

但是大多数现实的对话都是要多轮的，就是要结合上下文来做出回答。

很多研究者都意识到了上下文的重要性，也提出了很多方法。  
一种是直接将上下文和查询向量连接到一起。  
另一种是分层模型。  
有很多方法来结合上下文和查询，比如池化和连接。  
但是没有人对它们做过比较。

这篇论文里，我们在Seq2Seq的对话系统上对上下文模型做研究。

我们关注两个问题：

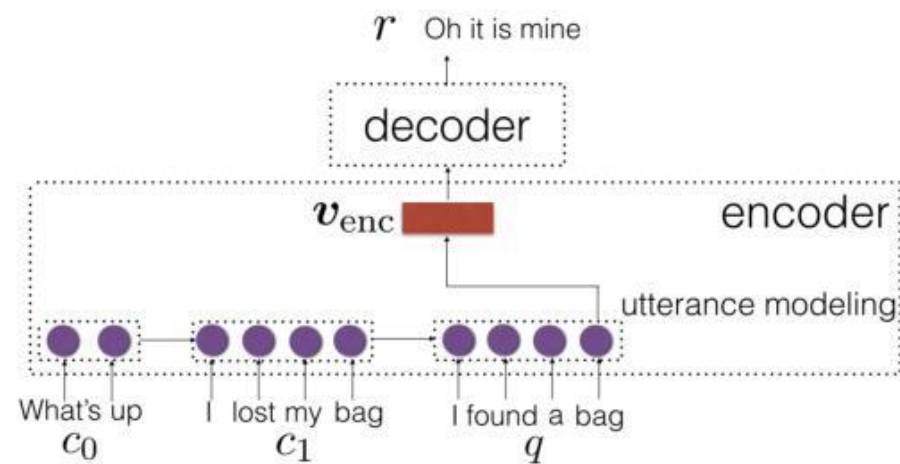
RQ1.我们怎么样才能更好的利用上下文？

RQ2.上下文对神经对话系统的影响是什么？

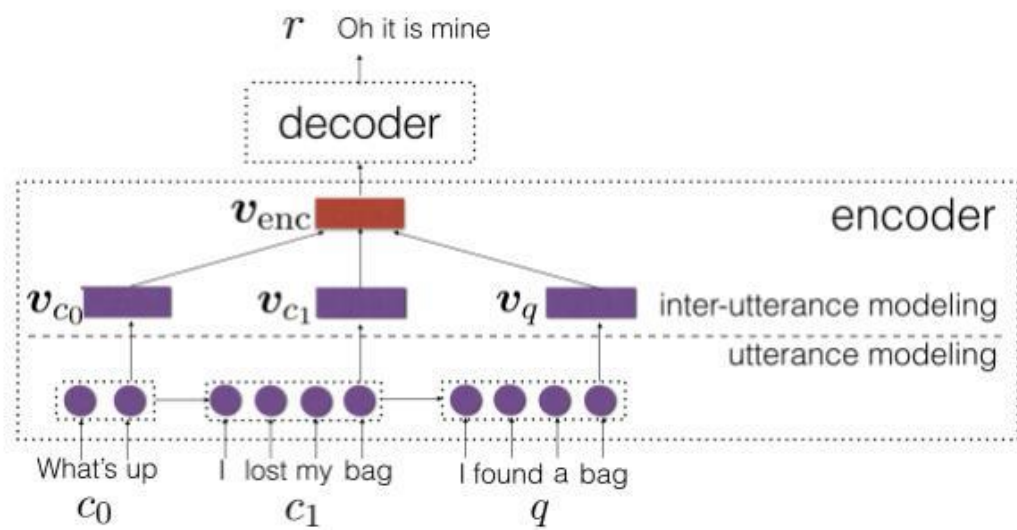
未分层模型通常使用经典的encode-decoder框架，我们实验中用的是RNN+GRU，decode时beam search大小为5。

分层模型有三种方法结合上下文和查询：

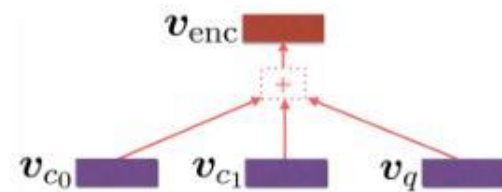
池化、连接、连续整合



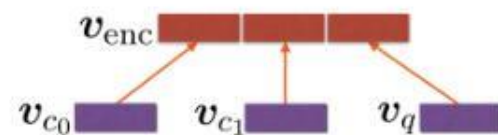
(a) Non-hierarchical model.



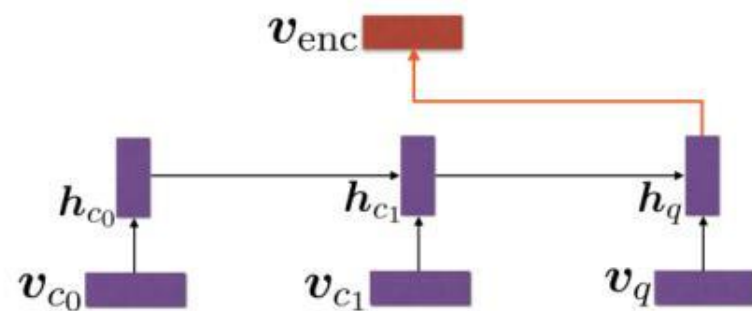
(b) Hierarchical model.



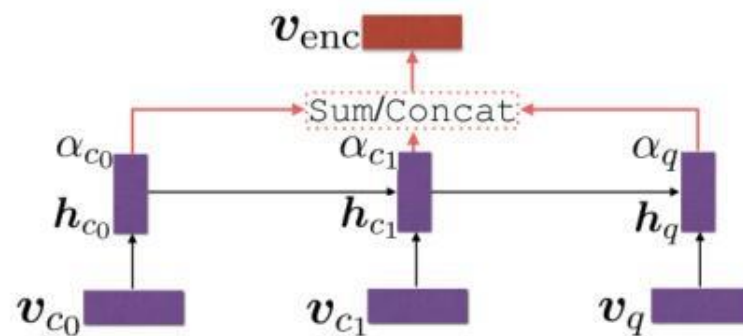
(a) Sum pooling.



(b) Concatenation.



(c) Sequential Integration.



(d) Weighted Sequential Integration.



但是我们发现加权后的实验结果还不如直接使用最后一个隐含层的结果，我们猜测是因为这个RNN不是很长，所以对前面的结果保存的比较好，所以我们实验直接使用最后一个隐含层作为输出。

衡量上下文和查询的相关程度：

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

where

$$\mathbf{e}_{c_i} = \sum_{w \in c_i} \mathbf{e}_w \quad \text{and} \quad \mathbf{e}_q = \sum_{w' \in q} \mathbf{e}_{w'} \quad (3)$$

归一化:

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

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

两种连接方法:

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

$$\mathbf{v}_{\text{enc}} = \sum_{i=0}^n \alpha_{c_i} \mathbf{h}_{c_i} + \alpha_q \mathbf{h}_q \quad (6)$$

- WSeq (concat), where weighted vectors are concatenated

$$\mathbf{v}_{\text{enc}} = [\alpha_{c_0} \mathbf{h}_{c_0}; \dots; \alpha_{c_n} \mathbf{h}_{c_n}; \alpha_q \mathbf{h}_q] \quad (7)$$

# 实验结果

我们在百度贴吧问答数据集上做实验。

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	<b>5.355</b>	1.771	0.916	0.387
	WSeq (sum)	5.134	1.586	0.7429	0.4359
	WSeq (concat)	5.322	<b>1.883</b>	<b>0.9966</b>	<b>0.6897</b>

Table 1: Performance of different models.

# 实验结果分析

虽然BLEU不适合用来度量对话系统这种开放式的系统。但是我们没有足够的人力物力来对结果一一标注，所以还是采用BLEU。

下面回到最开始提出的两个问题。

# 我们怎么样才能更好的利用上下文？

首先我们发现采用上下文实验结果的确比不采用的更好了。

然后分层的模型结果比不分层的更好。我们猜测原因可能是对话系统不同于其他NLP任务，对话系统句子可能出自不同的人。

让每个上下文保持独立很重要，而不是简单的池化结合到一起，所以直接连接起来效果更好。

上下文和查询相关性对系统有帮助。

# 上下文对神经对话系统的影响是什么？

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.



可以看出，采用上下文的模型能产生更长、更有意义、更多样性的回答。

我们还发现了一个有趣的现象：一个encode-decoder模型如果想要生成有意义的回答，必须要足够多的有意义的信息提供给它。这解释了为什么seq2seq在其他NLP任务表现得很好，但是在对话系统表现得不好。