Neural Responding Machine for Short-Text Conversation

Lifeng Shang, Zhengdong Lu, Hang Li
Noah's Ark Lab
Huawei Technologies Co. Ltd.
Sha Tin, Hong Kong
{shang.lifeng,lu.zhengdong,hangli.hl}@huawei.com

Abstract

We propose Neural Responding Machine (NRM), a neural network-based response generator for Short-Text Conversation. NRM takes the general encoder-decoder framework: it formalizes the generation of response as a decoding process based on the latent representation of the input text, while both encoding and decoding are realized with recurrent neural networks (RNN). The NRM is trained with a large amount of one-round conversation data collected from a microblogging service. Empirical study shows that NRM can generate grammatically correct and content-wise appropriate responses to over 75% of the input text, outperforming state-of-the-arts in the same setting, including retrieval-based and SMT-based models.

1 Introduction

Natural language conversation is one of the most challenging artificial intelligence problems, which involves language understanding, reasoning, and the utilization of common sense knowledge. Previous works in this direction mainly focus on either rule-based or learning-based methods (Williams and Young, 2007; Schatzmann et al., 2006; Misu et al., 2012; Litman et al., 2000). These types of methods often rely on manual effort in designing rules or automatic training of model with a particular learning algorithm and a small amount of data, which makes it difficult to develop an extensible open domain conversation system.

Recently due to the explosive growth of microblogging services such as Twitter[1] and Weibo[2], the amount of conversation data available on the web has tremendously increased. This makes a data-driven approach to attack the conversation problem (Ji et al., 2014; Ritter et al., 2011) possible. Instead of multiple rounds of conversation, the task at hand, referred to as Short-Text Conversation (STC), only considers one round of conversation, in which each round is formed by two short texts, with the former being an input (referred to as post) from a user and the latter a response given by the computer. The research on STC may shed light on understanding the complicated mechanism of natural language conversation.

Previous methods for STC fall into two categories, 1) the retrieval-based method (Ji et al., 2014), and 2) the statistical machine translation (SMT) based method (Sordoni et al., 2015; Ritter et al., 2011). The basic idea of retrieval-based method is to pick a suitable response by ranking the candidate responses with a linear or non-linear combination of various matching features (e.g. number of shared words). The main drawbacks of the retrieval-based method are the following

- the responses are pre-existing and hard to customize for the particular text or requirement from the task, e.g., style or attitude.
- the use of matching features alone is usually not sufficient for distinguishing positive responses from negative ones, even after time consuming feature engineering. (e.g., a penalty due to mismatched named entities is difficult to incorporate into the model)

The SMT-based method, on the other hand, is generative. Basically it treats the response generation as a translation problem, in which the model is trained on a parallel corpus of post-response pairs. Despite its generative nature, the method is intrinsically unsuitable for response generation, because the responses are not semantically equivalent to the posts as in translation. Actually one post can receive responses with completely different content, as manifested through the example in the fol-

[1] https://twitter.com/.

[2] http://www.weibo.com/.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics
and the 7th International Joint Conference on Natural Language Processing, pages 1577–1586,
Beijing, China, July 26-31, 2015.

lowing figure:

Post	Having my fish sandwich right now
UserA	For god's sake, it is 11 in the morning
UserB	Enhhhh... sounds yummy
UserC	which restaurant exactly?

Empirical studies also showed that SMT-based methods often yield responses with grammatical errors and in rigid forms, due to the unnecessary alignment between the "source" post and the "target" response (Ritter et al., 2011). This rigidity is still a serious problem in the recent work of (Sordoni et al., 2015), despite its use of neural network-based generative model as features in decoding.

1.1 Overview

In this paper, we take a probabilistic model to address the response generation problem, and propose employing a neural encoder-decoder for this task, named *Neural Responding Machine* (NRM). The neural encoder-decoder model, as illustrated in Figure 1, first summarizes the post as a vector representation, then feeds this representation to a decoder to generate responses. We further generalize this scheme to allow the post representation to dynamically change during the generation process, following the idea in (Bahdanau et al., 2014) originally proposed for neural-network-based machine translation with automatic alignment.

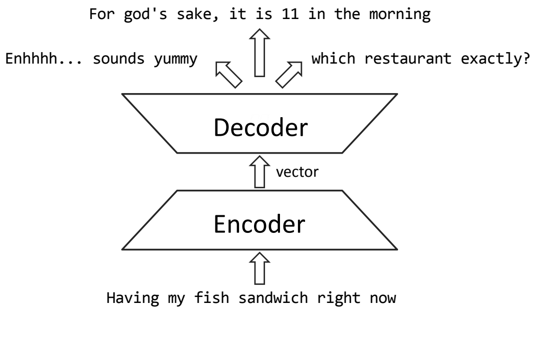
For god's sake, it is 11 in the morning
Enhhhh... sounds yummy
which restaurant exactly?
Decoder
vector
Encoder
Having my fish sandwich right now

Figure 1: The diagram of encoder-decoder framework for automatic response generation.

NRM essentially estimates the likelihood of a response given a post. Clearly the estimated probability should be complex enough to represent all the suitable responses. Similar framework has been used for machine translation with a remarkable success (Kalchbrenner and Blunsom, 2013; Auli et al., 2013; Sutskever et al., 2014; Bahdanau et al., 2014). Note that in machine translation, the task is to estimate the probability of a target language sentence conditioned on the source language sentence with the same meaning, which is much easier than the task of STC which we are considering here. In this paper, we demonstrate that NRM, when equipped with a reasonable amount of data, can yield a satisfying estimator of responses (hence response generator) for STC, despite the difficulty of the task.

Our main contributions are two-folds: 1) we propose to use an encoder-decoder-based neural network to generate a response in STC; 2) we have empirically verified that the proposed method, when trained with a reasonable amount of data, can yield performance better than traditional retrieval-based and translation-based methods.

1.2 RoadMap

In the remainder of this paper, we start with introducing the dataset for STC in Section 2. Then we elaborate on the model of NRM in Section 3, followed by the details on training in Section 4. After that, we report the experimental results in Section 5. In Section 6 we conclude the paper.

2 The Dataset for STC

Our models are trained on a corpus of roughly 4.4 million pairs of conversations from Weibo [3].

2.1 Conversations on Sina Weibo

Weibo is a popular Twitter-like microblogging service in China, on which a user can post short messages (referred to as *post* in the reminder of this paper) visible to the public or a group of users following her/him. Other users make comment on a published post, which will be referred to as a *response*. Just like Twitter, Weibo also has the length limit of 140 Chinese characters on both posts and responses, making the post-response pair an ideal surrogate for short-text conversation.

2.2 Dataset Description

To construct this million scale dataset, we first crawl hundreds of millions of post-response pairs, and then clean the raw data in a similar way as suggested in (Wang et al., 2013), including 1) removing trivial responses like "wow", 2) filtering out potential advertisements, and 3) removing the responses after first 30 ones for topic consistency. Table 1 shows some statistics of the dataset used

[3] http://www.noahlab.com.hk/topics/ShortTextConversation

Training	#posts	219,905
	#responses	4,308,211
	#pairs	4,435,959
Test Data	#test posts	110
Labeled Dataset (retrieval-based)	#posts	225
	#responses	6,017
	#labeled pairs	6,017
Fine Tuning (SMT-based)	#posts	2,925
	#responses	3,000
	#pairs	3,000

Table 1: Some statistics of the dataset. **Labeled Dataset** and **Fine Tuning** are used by retrieval-based method for learning to rank and SMT-based method for fine tuning, respectively.

in this work. It can be seen that each post have 20 different responses on average. In addition to the semantic gap between post and its responses, this is another key difference to a general parallel data set used for traditional translation.

3 Neural Responding Machines for STC

The basic idea of NRM is to build a hidden representation of a post, and then generate the response based on it, as shown in Figure 2. In the particular illustration, the encoder converts the input sequence $\mathbf{x} = (x_1, \cdots, x_T)$ into a set of high-dimensional hidden representations $\mathbf{h} = (h_1, \cdots, h_T)$, which, along with the attention signal at time t (denoted as α_t), are fed to the context-generator to build the context input to decoder at time t (denoted as c_t). Then c_t is linearly transformed by a matrix $\mathbf{L}$ (as part of the decoder) into a stimulus of generating RNN to produce the t-th word of response (denoted as y_t).

In neural translation system, $\mathbf{L}$ converts the representation in source language to that of target language. In NRM, $\mathbf{L}$ plays a more difficult role: it needs to transform the representation of post (or some part of it) to the rich representation of many plausible responses. It is a bit surprising that this can be achieved to a reasonable level with a linear transformation in the "space of representation", as validated in Section 5.3, where we show that one post can actually invoke many different responses from NRM.

The role of attention signal is to determine which part of the hidden representation $\mathbf{h}$ should be emphasized during the generation process. It should be noted that α_t could be fixed over time or changes dynamically during the generation of response sequence $\mathbf{y}$. In the dynamic settings, α_t can be function of historically generated subsequence $(y_1, \cdots, y_{t-1})$, input sequence $\mathbf{x}$ or their latent representations, more details will be shown later in Section 3.2.

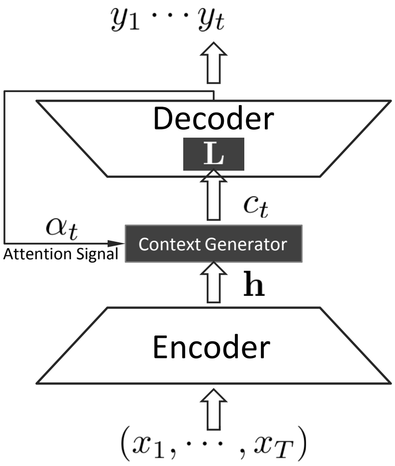

Figure 2: The general framework and dataflow of the encoder-decoder-based NRM.

We use Recurrent Neural Network (RNN) for both encoder and decoder, for its natural ability to summarize and generate word sequence of arbitrary lengths (Mikolov et al., 2010; Sutskever et al., 2014; Cho et al., 2014).

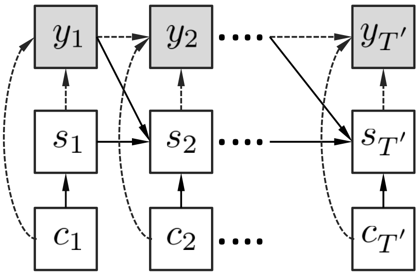

Figure 3: The graphical model of RNN decoder. The dashed lines denote the variables related to the function $g(\cdot)$, and the solid lines denote the variables related to the function $f(\cdot)$.

3.1 The Computation in Decoder

Figure 3 gives the graphical model of the decoder, which is essentially a standard RNN language model except conditioned on the context input $\mathbf{c}$. The generation probability of the t-th word is calculated by

$$p(y_t | y_{t-1}, \cdots, y_1, \mathbf{x}) = g(y_{t-1}, s_t, c_t),$$

where y_t is a one-hot word representation, $g(\cdot)$ is a softmax activation function, and s_t is the hidden state of decoder at time t calculated by

$$s_t = f(y_{t-1}, s_{t-1}, c_t),$$

and $f(\cdot)$ is a non-linear activation function and the transformation $\mathbf{L}$ is often assigned as parameters of $f(\cdot)$. Here $f(\cdot)$ can be a logistic function, the sophisticated long short-term memory (LSTM) unit (Hochreiter and Schmidhuber, 1997), or the recently proposed gated recurrent unit (GRU) (Chung et al., 2014; Cho et al., 2014). Compared to "ungated" logistic function, LSTM and GRU are specially designed for its long term memory: it can store information over extended time steps without too much decay. We use GRU in this work, since it performs comparably to LSTM on squence modeling (Chung et al., 2014; Greff et al., 2015), but has less parameters and easier to train.

We adopt the notation of GRU from (Bahdanau et al., 2014), the hidden state s_t at time t is a linear combination of its previous hidden state s_{t-1} and a new candidate state $\hat{s}_t$:

$$s_t = (1 - z_t) \circ s_{t-1} + z_t \circ \hat{s}_t,$$

where $\circ$ is point-wise multiplication, z_t is the update gate calculated by

$$z_t = \sigma\left(W_z e(y_{t-1}) + U_z s_{t-1} + L_z c_t\right), \quad (1)$$

and $\hat{s}_t$ is calculated by

$$\hat{s}_t = \tanh\left(W e(y_{t-1}) + U(r_t \circ s_{t-1}) + L c_t\right), \quad (2)$$

where the reset gate r_t is calculated by

$$r_t = \sigma\left(W_r e(y_{t-1}) + U_r s_{t-1} + L_r c_t\right). \quad (3)$$

In Equation (1)-(2), and (3), $e(y_{t-1})$ is word embedding of the word y_{t-1}, $\mathbf{L} = \{L, L_z, L_r\}$ specifies the transformations to convert a hidden representation from encoder to that of decoder. In the STC task, $\mathbf{L}$ should have the ability to transform one post (or its segments) to multiple different words of appropriate responses.

3.2 The Computation in Encoder

We consider three types of encoding schemes, namely 1) the global scheme, 2) the local scheme, and the hybrid scheme which combines 1) and 2).

Global Scheme: Figure 4 shows the graphical model of the RNN-encoder and related context generator for a global encoding scheme. The hidden state at time t is calculated by $h_t = f(x_t, h_{t-1})$ (i.e. still GRU unit), and with a trivial context generation operation, we essentially use the final hidden state h_T as the global representation of the sentence. The same strategy has been taken in (Cho et al., 2014) and (Sutskever et al., 2014) for building the intermediate representation for machine translation. This scheme however has its drawbacks: a vectorial summarization of the entire post is often hard to obtain and may lose important details for response generation, especially when the dimension of the hidden state is not big enough[4]. In the reminder of this paper, a NRM with this global encoding scheme is referred to as NRM-glo.

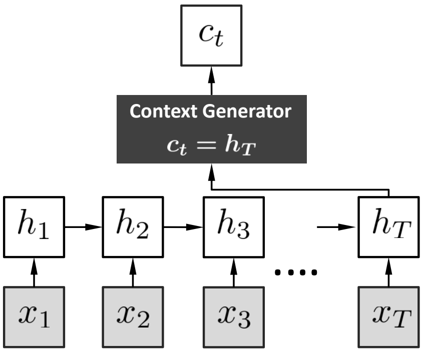

Figure 4: The graphical model of the encoder in NRM-glo, where the last hidden state is used as the context vector $c_t = h_T$.

Local Scheme: Recently, Bahdanau et al. (2014) and Graves (2013) introduced an attention mechanism that allows the decoder to dynamically select and linearly combine different parts of the input sequence $c_t = \sum_{j=1}^{T} \alpha_{tj} h_j$, where weighting factors α_{tj} determine which part should be selected to generate the new word y_t, which in turn is a function of hidden states $\alpha_{tj} = q(h_j, s_{t-1})$, as pictorially shown in Figure 5. Basically, the attention mechanism α_{tj} models the alignment between the inputs around position j and the output at position t, so it can be viewed as a local matching model. This local scheme is devised in (Bahdanau et al., 2014) for automatic alignment be-

[4] Sutskever et al. (2014) has to use $4,000$ dimension for satisfying performance on machine translation, while (Cho et al., 2014) with a smaller dimension perform poorly on translating an entire sentence.

tween the source sentence and the partial target sentence in machine translation. This scheme enjoys the advantage of adaptively focusing on some important words of the input text according to the generated words of response. A NRM with this local encoding scheme is referred to as NRM-loc.

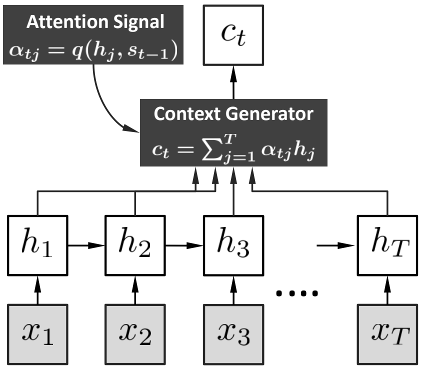

Figure 5: The graphical model of the encoder in NRM-loc, where the weighted sum of hidden sates is used as the context vector $c_t = \sum_{j=1}^{T} \alpha_{tj} h_j$.

3.3 Extensions: Local and Global Model

In the task of STC, NRM-glo has the summarization of the entire post, while NRM-loc can adaptively select the important words in post for various suitable responses. Since post-response pairs in STC are not strictly parallel and a word in different context can have different meanings, we conjecture that the global representation in NRM-glo may provide useful context for extracting the local context, therefore complementary to the scheme in NRM-loc. It is therefore a natural extension to combine the two models by concatenating their encoded hidden states to form an extended hidden representation for each time stamp, as illustrated in Figure 6. We can see the summarization h_T^g is incorporated into c_t and α_{tj} to provide a global context for local matching. With this hybrid method, we hope both the local and global information can be introduced into the generation of response. The model with this context generation mechanism is denoted as NRM-hyb.

It should be noticed that the context generator in NRM-hyb will evoke different encoding mechanisms in the global encoder and the local encoder, although they will be combined later in forming a unified representation. More specifically, the last hidden state of NRM-glo plays a role different from that of the last state of NRM-loc, since it has the responsibility to encode the entire input sentence. This role of NRM-glo, however, tends to be not adequately emphasized in training the hybrid encoder when the parameters of the two encoding RNNs are learned jointly from scratch. For this we use the following trick: we first initialize NRM-hyb with the parameters of NRM-loc and NRM-glo trained separately, then fine tune the parameters in encoder along with training the parameters of decoder.

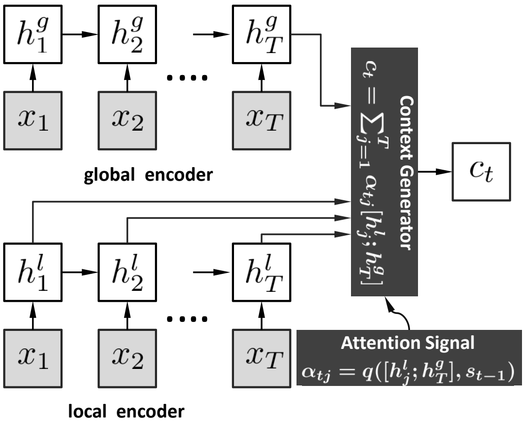

Figure 6: The graphical model for the encoder in NRM-hyb, while context generator function is $c_t = \sum_{j=1}^{T} \alpha_{tj}[h_j^l; h_T^g]$, here $[h_j^l; h_T^g]$ denotes the concatenation of vectors h_j^l and h_T^g

To learn the parameters of the model, we maximize the likelihood of observing the original response conditioned on the post in the training set. For a new post, NRMs generate their responses by using a left-to-right beam search with beam size = 10.

4 Experiments

We evaluate three different settings of NRM described in Section 3, namely NRM-glo, NRM-loc, and NRM-hyb, and compare them to retrieval-based and SMT-based methods.

4.1 Implementation Details

We use Stanford Chinese word segmenter [5] to split the posts and responses into sequences of words. Although both posts and responses are written in the same language, the distributions on words for the two are different: the number of unique words in post text is 125,237, and that of response text is 679,958. We therefore construct two separate vocabularies for posts and responses by using 40,000 most frequent words on each side, covering 97.8%

[5] http://nlp.stanford.edu/software/segmenter.shtml

usage of words for post and 96.2% for response respectively. All the remaining words are replaced by a special token "UNK". The dimensions of the hidden states of encoder and decoder are both 1,000. Model parameters are initialized by randomly sampling from a uniform distribution between -0.1 and 0.1. All our models were trained on a NVIDIA Tesla K40 GPU using stochastic gradient descent (SGD) algorithm with mini-batch. The training stage of each model took about two weeks.

4.2 Competitor Models

Retrieval-based: with retrieval-based models, for any given post p^*, the response r^* is retrieved from a big post-response pairs (p, r) repository. Such models rely on three key components: a big repository, sets of feature functions $\Phi_i(p^*, (p, r))$, and a machine learning model to combine these features. In this work, the whole 4.4 million Weibo pairs are used as the repository, 14 features, ranging from simple cosine similarity to some deep matching models (Ji et al., 2014) are used to determine the suitability of a post to a given post p^* through the following linear model

$$score(p^*, (p, r)) = \sum_i \omega_i \Phi_i(p^*, (p, r)). \quad (4)$$

Following the ranking strategy in (Ji et al., 2014), we pick 225 posts and about 30 retrieved responses for each of them given by a baseline retriever[6] from the 4.4M repository, and manually label them to obtain *labeled* 6,017 post-response pairs. We use ranking SVM model (Joachims, 2006) for the parameters ω_i based on the labeled dataset. In comparison to NRM, only the top one response is considered in the evaluation process.

SMT-based: In SMT-based models, the post-response pairs are directly used as parallel data for training a translation model. We use the most widely used open-source phrase-based translation model-Moses (Koehn et al., 2007). Another parallel data consisting of 3000 post-response pairs is used to tune the system. In (Ritter et al., 2011), the authors used a modified SMT model to obtain the "Response" of Twitter "Stimulus". The main modification is in replacing the standard GIZA++ word alignment model (Och and Ney, 2003) with a new phrase-pair selection method, in which all the possible phrase-pairs in the training data are considered and their associated probabilities are estimated by the Fisher's Exact Test, which yields performance slightly better than default setting[8]. Compared to retrieval-based methods, the generated responses by SMT-based methods often have fluency or even grammatical problems. In this work, we choose the Moses with default settings as our SMT model.

5 Results and Analysis

Automatic evaluation of response generation is still an open problem. The widely accepted evaluation methods in translation (e.g. BLEU score (Papineni et al., 2002)) do not apply, since the range of the suitable responses is so large that it is practically impossible to give reference with adequate coverage. It is also not reasonable to evaluate with Perplexity, a generally used measurement in statistical language modeling, because the naturalness of response and the relatedness to the post can not be well evaluated. We therefore resort to human judgement, similar to that taken in (Ritter et al., 2011) but with an important difference.

5.1 Evaluation Methods

We adopt human annotation to compare the performance of different models. Five labelers with at least three-year experience of Sina Weibo are invited to do human evaluation. Responses obtained from the five evaluated models are pooled and randomly permuted for each labeler. The labelers are instructed to imagine that they were the authors of the original posts and judge whether a response (generated or retrieved) is appropriate and natural to a input post. Three levels are assigned to a response with scores from 0 to 2:

- **Suitable (+2):** the response is evidently an appropriate and natural response to the post;
- **Neutral (+1):** the response can be a suitable response in a *specific* scenario;
- **Unsuitable (0):** it is hard or impossible to find a scenario where response is suitable.

To make the annotation task operable, the suitability of generated responses is judged from the following five criteria:

(a) **Grammar and Fluency**: Responses should be natural language and free of any fluency or grammatical errors;

[6] we use the default similarity function of Lucene [7]

[8] Reported results showed that the new model outperformed the baseline SMT model 57.7% of the time.

Post	意大利禁区里老是八个人...太夸张了 There are always 8 players at the Italian restricted area. Unbelievable!	**Related Criterion**	**Labels**
Response1	我是意大利队的球迷，等待比赛开始。 I am a big fan of the Italy team, waiting for the football game to start	Logic Consistency	Unsuitable
Response2	意大利的食物太美味了 Italian food is absolutely delicious.	Semantic Relevance	Unsuitable
Response3	太夸张了吧！ Unbelievable!	Generality	Neutral
Response4	哈哈哈仍然是0：0 。还没看到进球。 Ha!Ha!Ha! it is still 0:0, no goal so far.	Scenario Dependence	Neutral
Response5	这正是意大利式防守足球。 This is exactly the Italian defending style football game.	——	Suitable

Figure 7: An example post and its five candidate responses with human annotation. The content of the post implies that the football match is already started, while the author of Response1 is still waiting for the match to start. Response2 talks about the food of Italy. Response3 is a widely used response, but it is suitable to this post. Response4 states that the current score is still 0:0, it is a suitable response only in this specific scenario.

Models	**Mean Score**	**Suitable (+2)**	**Neutral (+1)**	**Unsuitable (0)**	**Agreement**
NRM-glo	0.969	34.0%	28.9%	37.1%	0.397
NRM-loc	1.065	36.0%	34.5%	29.5%	0.247
NRM-hyb	1.156	39.3%	37.1%	23.6%	0.309
Rtr.-based	0.984	29.8%	38.7%	31.5%	0.346
SMT-based	0.313	5.6 %	20.0%	74.4%	0.448

Table 2: The results of evaluated methods. Mean score is the average value of annotated scores over all annotations. (Rtr.-based means the retrieval-based method)

ID	**Post**	**NRM-glo**	**NRM-loc**	**NRM-hyb**	**Rtr.-based**
1	每年元旦都来高烧。 High fever attacks me every New Year's Day	多休息，增强免疫力！ Rest more to boost your immune system.	每年元旦都去医院看病 Go to see a doctor every New Year's Day	祝早日康复，身体健康！ Get well soon and stay healthy!	高烧… High fever…
2	又老了一岁，感谢我们组，我很高兴 I gain one more year. Grateful to my group, so happy	生日快乐！永远支持你！ Happy birthday! Will stand by you forever!	生日快乐！永远年轻漂亮！ Happy birthday! Wish you always stay young and pretty!	老了老了，岁月不饶人啊！ Getting old now. Time has no mercy.	我只是老了一岁 I just gain only one more year.
3	吃素第一天，坚持住，崔朵拉。 First day of being a vegetarian. Hold on, **CuiDuoLa**	必须来支持！加油！ I come to offer my support. Keep it up!	坚持住，坚持下去。 Hold on, keep it up.	每天都是素食主义者 I have been being a vegetarian everyday	坚持住！ Hold on!
4	这种天气真应该带个帅锅出去踏青赏花 We should go out with some cute guys to enjoy a great outing in such a nice weather.	这是哪儿啊？真美！ Where is it? It is so beautiful!	阳光明媚，心情舒畅。 Such a nice sunny day! I am in a great mood.	这种天气真不是盖的。 It is indeed a very nice weather.	文山啊 出去踏青寻找灵感 哈哈 **WenShan**, let's go out to get some inspiration. Ha! Ha!

Figure 8: Some responses generated by different models (originally in Chinese with their literal English translation), where the words in boldfaces are entity names.

(b) **Logic Consistency**: Responses should be logically consistent with the test post;

(c) **Semantic Relevance**: Responses should be semantically relevant to the test post;

(d) **Scenario Dependence**: Responses can depend on a specific scenario but should not contradict the first three criteria;

(e) **Generality**: Responses can be general but should not contradict the first three criteria;

If any of the first three criteria (a), (b), and (c) is contradicted, the generated response should be labeled as "Unsuitable". The responses that are general or suitable to post in a specific scenario should be labeled as "Neutral". Figure 7 shows an example of the labeling results of a post and its responses. The first two responses are labeled as "Unsuitable" because of the logic consistency and semantic relevance errors. Response4 depends on the scenario (i.e., the current score is 0:0), and is therefore annotated as "Neutral".

Model A	**Model B**	**Average rankings**	p **value**
NRM-loc	NRM-glo	(1.463, 1.537)	2.01%
NRM-hyb	NRM-glo	(1.434, 1.566)	0.01%
NRM-hyb	NRM-loc	(1.465, 1.535)	3.09%
Rtr.-based	NRM-glo	(1.512, 1.488)	48.1%
Rtr.-based	NRM-loc	(1.533, 1.467)	6.20%
Rtr.-based	**NRM-hyb**	(1.552, 1.448)	0.32%
SMT	**NRM-hyb**	(1.785, 1.215)	0.00 %
SMT	**Rtr.-based**	(1.738, 1.262)	0.00 %

Table 3: p-values and average rankings of Friedman test for pairwise model comparison. (Rtr.-based means the retrieval-based method)

5.2 Results

Our test set consists of 110 posts that do not appear in the training set, with length between 6 to 22 Chinese words and 12.5 words on average. The experimental results based on human annotation are summarized in Table 2, consisting of the ratio of three categories and the agreement among the five labelers for each model. The agreement is evaluated by Fleiss' kappa (Fleiss, 1971), as a statistical measure of inter-rater consistency. Except the SMT-based model, the value of agreement is in a range from 0.2 to 0.4 for all the other models, which should be interpreted as "Fair agreement". The SMT-based model has a relatively higher kappa value 0.448, which is larger than 0.4 and considered as "Moderate agreement", since the responses generated by the SMT often have the fluency and grammatical errors, making it easy to reach an agreement on such unsuitable cases.

From Table 2, we can see the SMT method performs significantly worse than the retrieval-based and NRM models and 74.4% of the generated responses were labeled as unsuitable mainly due to fluency and relevance errors. This observation confirms with our intuition that the STC dataset, with one post potentially corresponding to many responses, can not be simply taken as parallel corpus in a SMT model. Surprisingly, more than 60% of responses generated by all the three NRM are labeled as "Suitable" or "Neutral", which means that most generated responses are fluent and semantically relevant to post. Among all the NRM variants

- NRM-loc outperforms NRM-glo, suggesting that a dynamically generated context might be more effective than a "static" fixed-length vector for the entire post, which is consistent with the observation made in (Bahdanau et al., 2014) for machine translation;
- NRM-hyp outperforms NRM-loc and NRM-glo, suggesting that a global representation of post is complementary to dynamically generated local context.

The retrieval-based model has the similar mean score as NRM-glo, and its ratio on neutral cases outperforms all the other methods. This is because 1) the responses retrieved by retrieval-based method are actually written by human, so they do not suffer from grammatical and fluency problems, and 2) the combination of various feature functions potentially makes sure the picked responses are semantically relevant to test posts. However the picked responses are not customized for new test posts, so the ratio of suitable cases is lower than the three neural generation models.

To test statistical significance, we use the Friedman test (Howell, 2010), which is a non-parametric test on the differences of several related samples, based on ranking. Table 3 shows the average rankings over all annotations and the corresponding p-values for comparisons between different pairs of methods. The comparison between retrieval-based and NRM-glo is not significant and their difference in ranking is tiny. This indicates that the retrieval-based method is comparable to the NRM-glo method. The NRM-hyb outperforms all the other methods, and the difference is statistically significant ($p < 0.05$). The difference between NRM-loc and retrieval-based method is marginal ($p = 0.062$). SMT is significantly worse than retrieval-based and NRM-hyb methods.

5.3 Case Study

Figure 8 shows some example responses generated by our NRMs (only the one with biggest likelihood is given) and the comparable retrieval-based model. It is intriguing to notice that three NRM variants give suitable but quite distinct responses, with different perspectives and choices of words. This, as we conjecture, is caused by both the architecture variations among models as well as the variations from random effects like the initialization of parameters. Another interesting observation is on the fourth example, where the retrieval-based method returns a response with the mismatched entity name "WenShan", which is actually a quite common problem for retrieval-based model, where the inconsistency details (e.g., dates, named entities), which often render the response unsuitable, cannot be adequately considered in the matching function employed in retrieving the responses. In contrast, we observe that NRMs tend to make general response and barely generate those details.

Post	吃素第一天，坚持住，崔朵拉。 First day of being a vegetarian. Hold on, **CuiDuoLa**
R1	坚持住，坚持素食！ Hold on, Keep up your vegetarian diet.
R2	想要健康吗？想要长寿吗？那就关注我的微博吧。 Would you like to be healthy? Would you like to live long? If so, follow me on Weibo.
R3	这就是传说中的吃素吗？ This is the so-called vegetarian diet ?
R4	吃饱了才有力气减肥 Eat up so that you will have enough energy to go for a diet!
R5	身体是革命的本钱，不要着急！ Health is the essence for winning your future battles. No rush!

Figure 9: Multiple responses generated by the NRM-hyb.

We also use the NRM-hyb as an example to investigate the ability of NRM to generate multiple responses. Figure 9 lists 5 responses to the same post, which are gotten with beam search with beam size = 500, among which we keep only the best one (biggest likelihood) for each first word. It can be seen that the responses are fluent, relevant to the post, and still vastly different from each other, validating our initial conjecture that NRM, when fueled with large and rich training corpus, could work as a generator that can cover a lot of modes in its density estimation.

It is worth mentioning that automatic evaluation metrics, such as BLEU (Papineni et al., 2002) as adopted by machine translation and recently SMT-based responding models (Sordoni et al., 2015), do not work very well on this task, especially when the reference responses are few. Our results show that the average BLEU values are less than 2 for all models discussed in this paper, including SMT-based ones, on instances with single reference. Probably more importantly, the ranking given by the BLEU value diverges greatly from the human judgment of response quality.

6 Conclusions and Future Work

In this paper, we explored using encoder-decoder-based neural network system, with coined name Neural Responding Machine, to generate responses to a post. Empirical studies confirm that the newly proposed NRMs, especially the hybrid encoding scheme, can outperform state-of-the-art retrieval-based and SMT-based methods. Our preliminary study also shows that NRM can generate multiple responses with great variety to a given post. In future work, we would consider adding the intention (or sentiment) of users as an external signal of decoder to generate responses with specific goals.

Acknowledgments

The authors would like to thank Tao Cai for technical support. This work is supported in part by China National 973 project 2014CB340301.

References

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. 2013. Joint language and translation modeling with recurrent neural networks. In *EMNLP*, pages 1044–1054.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *preprint arXiv:1308.0850*.

Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2015. LSTM: A search space odyssey. *CoRR*, abs/1503.04069.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

David C. Howell. 2010. *Fundamental Statistics for the Behavioral Sciences*. PSY 200 (300) Quantitative Methods in Psychology Series. Wadsworth Cengage Learning.

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

Thorsten Joachims. 2006. Training linear svms in linear time. In *SIGKDD*, pages 217–226. ACM.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP*, pages 1700–1709.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. ACL.

Diane Litman, Satinder Singh, Michael Kearns, and Marilyn Walker. 2000. Njfun: a reinforcement learning spoken dialogue system. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems*, pages 17–20. ACL.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010*, pages 1045–1048.

Teruhisa Misu, Kallirroi Georgila, Anton Leuski, and David Traum. 2012. Reinforcement learning of question-answering dialogue policies for virtual museum guides. In *SIGDIAL*, pages 84–93. ACL.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *EMNLP*, pages 583–593. Association for Computational Linguistics.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(02):97–126.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Meg Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies (NAACL-HLT 2015), June.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Hao Wang, Zhengdong Lu, Hang Li, and Enhong Chen. 2013. A dataset for research on short-text conversations. In *EMNLP*, pages 935–945.

Jason D Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.