Neural Utterance Ranking Model for Conversational Dialogue Systems

Michimasa Inaba
Hiroshima City University
3-4-1 Ozukahigashi, Asaminami-ku,
Hiroshima, Japan
inaba@hiroshima-cu.ac.jp

Kenichi Takahashi
Hiroshima City University
3-4-1 Ozukahigashi, Asaminami-ku,
Hiroshima, Japan
takahashi@hiroshima-cu.ac.jp

Abstract

In this study, we present our neural utterance ranking (NUR) model, an utterance selection model for conversational dialogue agents. The NUR model ranks candidate utterances with respect to their suitability in relation to a given context using neural networks; in addition, a dialogue system based on the model converses with humans using highly ranked utterances. Specifically, the model processes word sequences in utterances and utterance sequences in context via recurrent neural networks. Experimental results show that the proposed model ranks utterances with higher precision relative to deep learning and other existing methods. Furthermore, we construct a conversational dialogue system based on the proposed method and conduct experiments on human subjects to evaluate performance. The experimental result indicates that our system can offer a response that does not provoke a critical dialogue breakdown with a probability of 92% and a very natural response with a probability of 58%.

1 Introduction

The study of conversational dialogue systems (also known as non-task-oriented or chat-oriented dialogue systems) has a long history. To construct such systems, rule-based methods have long been used (Weizenbaum, 1966; Colby, 1981; Wallace, 2008); however, construction and maintenance costs are very high because these rules are manually created. Moreover, intuition tells us that the performance of such systems depends on the number of established rules, though reports indicate that performance did not improve much even if the number of rules was doubled (Higashinaka et al., 2015b), indicating that performance of rule-based systems is limited.

Recently, the study of statistical-based methods that use statistical processing with large volumes of web data has become increasingly active. The key benefit of this approach is that manual response creation is not necessary; thus, construction and maintenance costs are low; however, since web data contains noise, this approach has the potential to output grammatically or semantically incorrect sentences. To tackle this problem, some studies extract correct sentences as utterances for dialogue systems from web data (Inaba et al., 2014; Higashinaka et al., 2014). These studies focus solely on extraction and do not indicate how replies are generated using extracted sentences.

In our study, we propose a neural utterance ranking (NUR) model that ranks candidate utterances by their suitability in a given context using neural networks. Previously, we proposed an utterance selection model (Koshinda et al., 2015) in the framework same as that of the NUR model, which ranks utterances in order of suitability to given context. In section 4, we experimentally show that the performance of the NUR model exceeds that of our previous model.

Our proposed method processes the word sequences in utterances and utterance sequences in context via multiple recurrent neural networks (RNNs). More specifically, the RNN encodes both utterances in a given context and candidates into fixed-length vectors. Such encoding enables suitable feature extraction for ranking. Next, another RNN receives these utterance-encoded vectors in chronological order, and our proposed NUR model ranks candidates using the output of this RNN. Our model considers the order of utterances in a given context; this architecture makes it pos-

Proceedings of the SIGDIAL 2016 Conference, pages 393–403,
Los Angeles, USA, 13-15 September 2016. ©2016 Association for Computational Linguistics

sible to handle distant semantic relationships between context and candidates.

2 Related Work

Statistical-based response methods incorporate two major approaches.

The first approach is the example-based method (Murao et al., 2003), which searches a large database of previously recorded dialogue for given user input selecting an utterance identified as the most similar. Well-known dialogue systems based on this approach include Jabberwacky (De Angeli and Carpenter, 2005) which won the Loebner prize contest[a] (i.e., a conversational dialogue system competition) in 2005 and 2006. In addition, Banch and Li. proposed a model based on the vector space model (Banchs and Li, 2012) and Nio et al. constructed a dialogue system that uses movie scripts and Twitter data (Nio et al., 2014). A disadvantage of example-based methods is that it is difficult to consider context. If the implemented approach searches for user input with context in a database, it can be difficult to find a suitable context because of the diversity of contexts; in such cases, system replies become unsuitable. In contrast, our NUR model can select responses while also taking into account a flexible set of contexts.

The second statistical-based response approach is the machine translation (MT) method. Ritter et al. first introduced the MT technique into response generation (Ritter et al., 2011). They used tweet-reply pairs in Twitter data, regarding a tweet as the source language sentence and the reply as a target one in MT. In other words, the MT method translates user input into system responses. More recently, response generation using neural networks has been widely studied, most work grounded in the MT method (Cho et al., 2014; Sordoni et al., 2015; Shang et al., 2015; Vinyals and Le, 2015). A problem with this method is that it might generate utterances containing syntax errors; further, it tends to generate utterances with broad utility that frequently appear in training data, e.g., "I don't know." or "I'm OK." (Li et al., 2016).

Our proposed method is not categorized into either of the above two methods. Some hard-to-classify statistical-based response methods similar to our model have been proposed, e.g., Shibata et al. proposed a method that selects a suitable sentence extracted from webpages as a response to user input (Shibata et al., 2009). Sugiyama et al. generated responses using templates and dependency structures of sentences gathered from Twitter (Sugiyama et al., 2013). There are only few common points, although most of the hard-to-classify methods use not only dialogue data but also non-dialogue data such as webpages or normal tweets (not pairs of tweet reply) on Twitter.

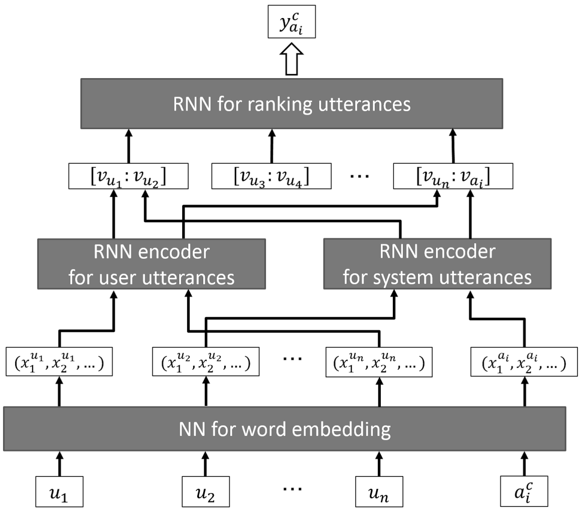
$y_{a_i}^c$
RNN for ranking utterances
$[v_{u_1}:v_{u_2}]$
$[v_{u_3}:v_{u_4}]$
$\cdots$
$[v_{u_n}:v_{a_i}]$
RNN encoder for user utterances
RNN encoder for system utterances
$(x_1^{u_1}, x_2^{u_1}, \ldots)$
$(x_1^{u_2}, x_2^{u_2}, \ldots)$
$\cdots$
$(x_1^{u_n}, x_2^{u_n}, \ldots)$
$(x_1^{a_i}, x_2^{a_i}, \ldots)$
NN for word embedding
u_1
u_2
$\cdots$
u_n
a_i^c

Figure 1: Neural utterance ranking model

3 Neural Utterance Ranking Model

For our ranking model, we first define sequences of utterances from the beginning of a dialogue to a certain point of time in context $c = (u_1, u_2, \ldots, u_l)$ Each $u_i (i = 1, 2, ..., l)$ denotes an utterance in the context, and l denotes the number of utterances. We assume here that a dialogue system and user speak alternately and last utterance u_l is given by the system. We define candidate utterance list $a_c = (a_1^c, a_2^c, \ldots, a_m^c)$ generated depending on context c, and score $t_c = (t_1^c, t_2^c, \ldots, t_m^c)$. Herein, m denotes the number of candidate utterances. We define utterance ranking to sort given candidate utterance list a_c in order of suitability to context c. The correct order is defined by score t_c with sorting based on the model's output $y_{a_c} = (y_1, y_2, \ldots, y_m)$ corresponding to a_c.

Our proposed utterance ranking model, i.e., the NUR model illustrated in in Figure 1, receives context c and candidate utterance list a_c, then outputs y_{a_c}. Details of our NUR model are described below.

[a] http://www.loebner.net/Prizef/loebner-prize.html

3.1 Utterance Encoding

To extract information from context and candidate utterances for suitable utterance selection, our NUR model utilizes an RNN encoder.

Previous work utilized an RNN encoder for MT (Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2015) and response generation in dialogue systems (Cho et al., 2014; Sordoni et al., 2015; Shang et al., 2015; Vinyals and Le, 2015). In these studies, the encoder reads as input a variable-length word sequence and outputs a fixed-length vector. Next, another RNN decodes a given fixed-length vector, producing an objective variable-length word sequence. Therefore, the encoder has learned to embed necessary information to generate objective sentences and place them into vectors. The RNN in our model does not generate sentences using this RNN decoder approach. Results of encoding are used for features to rank candidate utterances. The RNN encoder in our NUR model has a similar architecture, but the characteristics of the output vector are profoundly different, because our model learns to extract important features for utterance ranking.

Our model first converts word sequence $w = (w_1, w_2, \ldots, w_n)$ in an utterance into a distributed representation of word sequence, i.e., $x = (x_1, x_2, \ldots, x_n)$ which the RNN encoder then reads. To convert into a distributed representation here, a neural network for word embedding (as shown in Figure 1) learns via the skip-gram model (Mikolov et al., 2013). This network has two layers, i.e., an input layer that reads a one-hot-vector representing each word and a certain denominational hidden layer.

The RNN encoder has two networks, i.e., a forward and a backward network. The forward RNN reads x at the beginning of a sentence and outputs $\overrightarrow{h} = (\overrightarrow{h_1}, \overrightarrow{h_2}, \ldots, \overrightarrow{h_n})$ correspond to input sequence. The backward RNN reads x in reverse, then outputs $\overleftarrow{h} = (\overleftarrow{h_1}, \overleftarrow{h_2}, \ldots, \overleftarrow{h_n})$. By joining the outputs of these forward and backward RNNs, we acquire objective encoded utterance vector $v = [\overrightarrow{h_n}; \overleftarrow{h_n}]$; note that $[x; y]$the concatenation of vectors x and y.

In the following experiments, we used two-layer long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) networks as our RNN encoders. The effective features extracted from utterances for candidate ranking are different between the user and the system. Therefore, our NUR model has two RNN encoders, one for user utterances, the other for system utterances, as illustrated in Figure 1

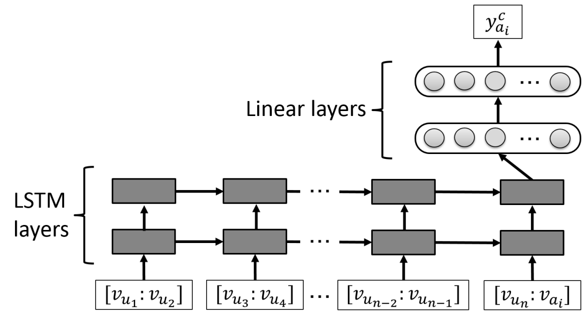

Figure 2: RNN for ranking utterances

3.2 Ranking Candidate Utterances

Another RNN is used to rank candidate utterances, as illustrated in 2. This RNN has two LSTM layers and two linear layers; further, we use rectified linear unit (ReLU) as the activation function. Thus, this RNN reads encoded utterance sequences and outputs scores.

3.2.1 Context-Candidate Vector Sequence

To select suitable responses, we not only must evaluate suitability of utterances based on the last utterance in the given context, but also must consider prior dialogue. The RNN for ranking utterances in our model reads vector sequences constructed by context and candidate utterances in chronological order, then outputs scores for the candidate in relation to the context.

Thus, context-candidate vector sequence $v^c_{a_i}$ is constructed using context vector sequence $v_c = (v_{u_1}, v_{u_2}, \ldots, v_{u_l})$, with ith candidate utterance vector v_{a_i} defined as follows:

$$v^c_{a_i} = \begin{cases} ([v_{u_1}; v_{u_2}], [v_{u_3}; v_{u_4}], \ldots, [v_{u_l}; v^c_{a_i}]), & \text{if } l \text{ is odd} \\ ([\mathbf{0}; v_{u_1}], [v_{u_2}; v_{u_3}], \ldots, [v_{u_l}; v^c_{a_i}]), & \text{if } l \text{ is even} \end{cases}$$

Here, $\mathbf{0}$ denotes the zero vector. Our model inputs user and system utterances at one time so that it can consider dialogue history in a given context along with the relevance between candidate utterances and the last response given by a user.

3.2.2 Loss Function in Learning

In cases where a neural network outputs a one-dimensional value, like our model, the mean squared error (MSE) between training data and the model's output is generally used as a loss function; however, our objective is not to model scores, but rather for ranking, thus we use the distance between rank data based on training data and that based on the model's outputs as a loss function. Several methods for modeling rank data have been proposed, including the Bradley-Terry-Luce model (Bradley and Terry, 1952; Luce, 1959), the Mallows model (Mallows, 1957) and the Plackett-Luce model (Plackett, 1975; Luce, 1959). In our study, to calculate ranking distance, we selected the Plackett-Luce model, which has been used in various ranking models, such as ListNet (Cao et al., 2007), BayesRank (Kuo et al., 2009), etc.

The Plackett-Luce model transforms a score list for ranking into a probability distribution wherein higher scores in the given list are allocated higher probabilities. Probability of score t_i^c in score list $t_c = (t_1^c, t_2^c, \ldots, t_m^c)$ ranked on the top is calculated by the Plackett-Luce model as follows:

$$p(t_i^c) = \frac{\exp(t_i^c)}{\sum_{k=1}^{m} \exp(t_k^c)}$$

Using the same equation, the output scores of our NUR model are transformed into probability distributions. We use cross-entropy between probability distributions as our loss function.

4 Experiments

We conducted experiments to verify the performance of ranking given candidate utterances and given contexts. For comparison, we also tested a few baseline methods.

4.1 Datasets

For our experiments, we used dialogue data between a conversational dialogue system and a user for both training and test data. We released a conversational dialogue system called KELDIC on Twitter (screen name: @KELDIC)[b]. KELDIC selects an appropriate response from candidates extracted by the utterance acquisition method of (Inaba et al., 2014) using ListNet(Cao et al., 2007). The utterance acquisition method extracted suitable sentences for system utterances related to given keywords from Twitter data by filtering inappropriate sentences. Details of the response algorithm of KELDIC is further described in (Koshinda et al., 2015).

We collected training and test data by first collecting pairs of context and candidate utterances that the system used for reply on Twitter. Next, annotators evaluated the suitability of each candidate utterance in relation to the given context. Here annotators must evaluate utterances that were actually used by the system on Twitter.

Evaluation criterion was based on the Dialogue Breakdown Detection Challenge (DBDC) (Higashinaka et al., 2016). Each system's utterances were annotated using one of the following three breakdown labels:

(NB) Not a breakdown It is easy to continue the conversation.

(PB) Possible breakdown It is difficult to continue the conversation smoothly.

(B) Breakdown It is difficult to continue the conversation.

Annotators evaluated dialogue data on a tool we prepared. They were first shown a context and 10 candidate utterances, including how KELDIC actually replied on Twitter, as well as labels for each candidate. We instructed them to assign at least one NB label to given candidate utterances. If there were no suitable candidates for the NB label, they could optionally add candidate utterances. If they were still not able to find a suitable response, we allowed them to skip the evaluation. We recruited annotators on crowd-sourcing site CrowdWorks[c].

In our evaluation, we regard candidates with 50% or more annotators decided as NB as correct utterances and others as incorrect.

We used 1581 data points (i.e., 1581 contexts and 17533 candidate utterances), each evaluated by three or more annotators. We choose 300 data points that contained at least one correct candidate for the given test data; the remaining 1057 data points were used for training data. Table 1 shows statistics for our data.

In learning the model, we need scores for candidate utterances to define ranking. Score y_i^c of candidate utterance a_i^c is calculated as follows:

$$y_i^c = s_{\mathrm{NB}} \frac{n_{\mathrm{NB}}}{N} + s_{\mathrm{PB}} \frac{n_{\mathrm{PB}}}{N} + s_{\mathrm{B}} \frac{n_{\mathrm{B}}}{N}$$

$$N = n_{\mathrm{NB}} + n_{\mathrm{PB}} + n_{\mathrm{B}}$$

[b] https://twitter.com/KELDIC

[c] https://crowdworks.jp

Table 1: Statistics of the datasets

	Train	Test	All
Data	1281	300	1581
Utterances in context	1.67	2.04	1.74
Candidates per data	11.12	10.94	11.09
Words per candidate	11.17	10.70	11.08
Num of Annotators	3.97	3.88	3.95

Here, n_{NB}, n_{PB} and n_{B} denote the numbers of annotators assigned as NB, PB and B, respectively, and s_{NB}, s_{PB} and s_{B} denote scoring parameters of NB, PB and B, respectively. In our experiments, we set $(s_{\mathrm{NB}}, s_{\mathrm{PB}}, s_{\mathrm{B}}) = (10.0, -5.0, -10.0)$.

4.2 Experimental Settings

In the word-embedding neural network of our NUR model, we used 1000 embedding cells, a skip-gram window size of five, and learned via 100GB of Twitter data (Other layers were learned by 1281 data points).

In our encoding and ranking RNNs, we used LSTM layers with 1000 hidden cells in each layer. The dropout rate was set to 0.5, and the model was trained via AdaGrad (Duchi et al., 2011).

To validate our NUR model, we conducted experiments with the following two settings:.

Proposed using limited context

To verify the effectiveness of context sequence processing by the ranking RNN, this setting causes our system to only use the last user utterance as context, discarding the rest.

Proposed using MSE

To verify the effectiveness of the Plackett-Luce model, this setting causes our system to learn using the MSE of utterance scores instead of the Plackett-Luce model.

We also compared performance to the following three methods:

BoW + DNN

This method ranks candidate utterances using deep neural networks (DNNs) and bag-of-words (BoW) features. The DNN consisted of six layers, excluding input and output layers optimized by MSE. The input vector is made by concatenating three BoW vectors, i.e., candidate utterance, last user utterance in the given context, and the given context without the last user utterance. In the BoW vector, we used 6203 words that occur at least two times in the training data, thus, the input layer of the DNN has 18609 cells. Each hidden layer has 5000 cells, with ReLU as the activation function, the dropout rate set to 0.5, and the model trained by AdaGrad (Duchi et al., 2011). The score for training was the same as the model proposed in Section 4.1.

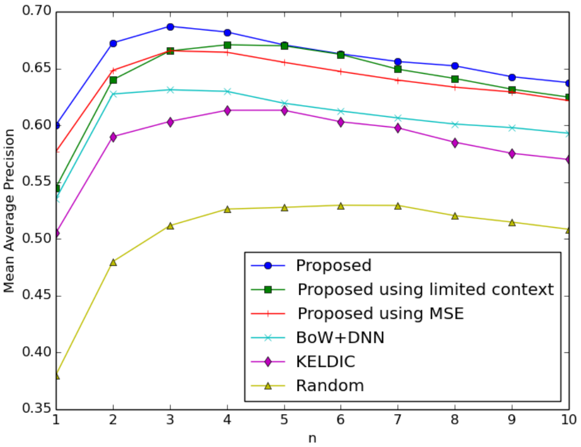

Figure 3: MAP over top n candidates

KELDIC

The second comparative approach used the output of our KELDIC system. This dialogue system ranks utterances using ListNet (Cao et al., 2007) and selects the top-ranked utterance to reply. The feature vector for ranking is generated from context and candidate utterance. It primarily utilizes n-gram pairs between utterances in context and candidates as features.

Random

This approach randomly shuffles candidates and uses them as a ranking list, thus serving as a baseline for ranking performance.

4.3 Results

To evaluate ranking performance, we used mean average precision (MAP) and mean reciprocal rank (MRR) measures. Figure 3 shows MAP results over the top n ranked candidate utterances, while Figure 4 shows MRR results. Using the MAP measure, our proposed method showed the highest performance as compared to the other methods. The proposed using limited context and MSE follow this, suggesting that utterance encoding by RNN is effective to extract features for ranking.

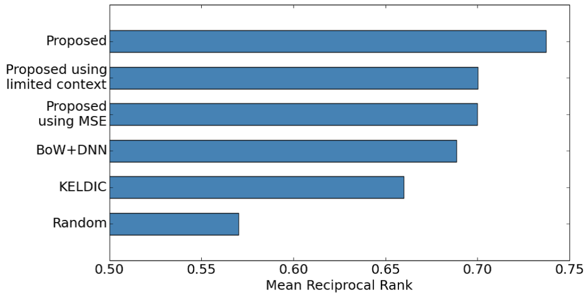

Figure 4: Mean Reciprocal Rank

BoW + DNN did not provide strong performance results, because it could not handle the order relation of utterances in context and syntax due to the use of BoW features. KELDIC showed higher performance than that of Random, but lower than that of BoW + DNN, because it also has the problem of context processing and its generalization capability is lower than that of DNNs.

Here, $n = 1$ of MAP indicates that the rate of correct utterance ranked on the top (The maximum value of $n = 1$ of MAP is 1.0 because each data points in the test data contains at least one correct candidate utterance). Since the top-ranked utterance is selected as a response in dialogue systems, it was found that our proposed method correctly replied with a probability of approximately 60%.

Results of MRR (i.e., Figure 4) showed very similar results, i.e., our proposed method ranked suitable utterances higher.

Table 2 shows an example of context in the test data and Table 3 shows candidate utterances to the context shown in Table 2, plus ranking results for the applied methods and NB rates of annotations for candidates. These results indicate that our proposed method ranked correct utterances higher and incorrect utterances lower, as desired.

5 Dialogue Experiment

In the previous section, since test data must contain correct candidate utterances, the ability of our NUR model in terms of actual dialogue is uncertain, thus we developed a conversational dialogue system based on our proposed method and conducted dialogue experiments with human subjects.

The dialogue format and rules were fully compliant with the Dialogue Breakdown Detection Challenge (DBDC) (see (Higashinaka et al., 2015a)). A dialogue is started by a system utterance, then user and the system communicate with one another. When a system speaks 11 times, the dialogue is finished. Therefore, a dialogue contains 11 system and 10 human utterances.

Our dialogue system and subjects chat on our website; we collected 120 text chat dialogues. Annotators then labeled 1200 system utterances (excluding initial greetings) using breakdown labels NB, PB, and B. We again recruited subjects and annotators via CrowdWorks.

For comparison, we used DBDC development/test data[d] collected by chatting with a system based on NTT Docomo's chat API[e] (see (Onishi and Yoshimura, 2014) . Since the DBDC system selects a suitable response from large-scale utterance data, the architecture is similar to our model and therefore suitable as a comparative system.

DBDC data has been annotated by 30 annotators using the breakdown labels and we use them without any change in this experiment. Therefore, the annotation rule is same but the annotators are different between our dialogue data and DBDC data.

5.1 Dialogue System

A conversational dialogue system based on our NUR model selects an utterance as a response from candidates generated by the acquisition method of (Inaba et al., 2014). The system extracts nouns from the last user and system utterances, generating candidate utterances related to nouns. We used approximately one billion Japanese tweets collected from January through February 2015 for utterance acquisition. Our NUR model ranked candidates, and the system used top-ranked utterances as responses. If there were less than five acquired utterances, the system retroactively extracted nouns in context one by one to acquire further candidates.

The first utterance in the beginning of a dialogue was randomly selected from 16 manually created open question utterances, such as "What is your favorite website?" or "What kind of part-time job do you have?". If the user's response does not contain any nouns or the number of acquired utterances is less than five, the system randomly selects the 16 utterances again.

5.2 Results

Table 4 shows statistics of the data, annotations, and experimental results. Dialogue data used in

[d] https://sites.google.com/site/dialoguebreakdowndetection/
[e] https://www.nttdocomo.co.jp/service/

Table 2: Example context (translated by authors)

Speaker	Utterance in context
User	大学行くモチベーションが見た目に如実に表れてる My motivation to attend college is cosmetically significant.
System	わかります。一夜が明け、仕事へのモチベーションはまずまずであるものの テンションは非常に低いよね I see. It's dawn and although I'm downhearted now, my motivation for working is not bad.
User	わかる Sure.
System	だよね！普段のテンションが高い分、落ち込んだらとことん落ち込むよ Indeed! I have high motivation on an average day, but once I get depressed, things become increasingly worse.
User	そうなんだ。落ち込むな元気出せ！ Oh, don't be so depressed and cheer up!

Table 3: Example candidate utterances with ranking results (translated by authors)

Proposed	BoW + DNN	KELDIC	NB	Candidate Utterance
1	6	3	1.00	相当なモチベーションが必要だよ It requires a considerable degree of motivation.
2	8	6	0.33	独学はモチベーションを保つのが大変だよ Self-education is difficult to keep me motivated.
3	10	2	0.00	どんなモチベーションでチャリこげばいいよね What is my motivation to pedal a bicycle?
4	1	8	0.33	勉強へのモチベーションがすごい My motivation to study is quite good.
5	9	4	0.33	モチベーションには繋がるよ It's to be a motivation.
6	4	9	0.00	ポケモンのモチベーションが皆無だよ I have no motivation to play Pokemon.
7	7	10	0.00	実習のモチベーション保つのって大変だね It's hard to stay motivated in practical training.
8	3	5	0.000	東方のモチベーションがすごくなってるよ My motivation to play Touhou games is quite high.
9	5	1	0.00	PC に対するモチベーション低いしやる気でない My motivation to use a PC is low, and I don't feel like doing anything.
10	2	7	0.00	モチベーション低い幹事は良くない An organizer who has low motivation is bad.

our system were annotated by 34 human annotators. Fleiss's K measure for our system's data was lower than that of the DBDC dataset, but both are low. "PB + B" indicates that PB and B are treated as a single label. The table also shows the ratio of NB, PB, and B labels. These annotation results indicate that output probabilities of PB and B utterances by our system were significantly lower, while NB was higher than that of the DBDC system ($p < 0.01$).

The Breakdown ratio (B) and (PB + B) values are calculated by the labels of majority vote in 34 (proposed) or 30 (DBDC) annotators in each system's utterance. Breakdown ratio (B) is the ratio of the B majority label to all majority labels. Breakdown ratio (PB + B) is the ratio of PB and B majority labels (treated as a single label). This indicates that our system can offer a response that does not provoke a critical dialogue breakdown with a probability of approximately 90% and a

Table 4: Statistics of the data and experimental results (U and S indicate statistics of user and system utterances, respectively)

	Proposed	DBDC
Dialogues	120	100
Utterances (U)	1200	1000
Utterances (S)	1320	1100
Words per utterance (U)	9.32	9.43
Words per utterance (S)	8.63	7.17
Vocabularies (U)	1684[f]	1491
Vocabularies (S)	1386[f]	1218
Annotators	34	30
NB (Not a breakdown)	57.7%	37.1%
PB (Possible breakdown)	27.0%	32.2%
B (Breakdown)	15.2 %	30.6%
Fleiss's κ (NB, PB, B)	0.26	0.20
Fleiss's κ (NB, PB+B)	0.37	0.27
Breakdown ratio (B)	0.08	0.25
Breakdown ratio (PB+B)	0.42	0.71

very natural response with a probability of 60%. Both breakdown ratios showed significant differences between our system and the DBDC system ($p < 0.001$).

Table 4 also shows the number of words per utterance and the number of vocabularies. These results are important for system evaluation, because if a system always use innocuous responses, such as "I don't know" or "That's true", it is relatively easy to avoid dialogue breakdown. By these values, we can find whether a system frequently uses innocuous responses or not; however, to increase user satisfaction with a dialogue system, it is important not only to avoid dialogue breakdown, but also to offer flexible replies. From Table 4, we also observe that the number of words per utterance and the number of vocabularies in our system were bigger than that of the DBDC system, indicating that our system infrequently used innocuous responses and had a good vocabulary for generating responses. Indeed, our system rarely used such utterances, but the DBDC system sometimes used them.

The number of words per utterance by user between both datasets was almost the same, but the number of vocabularies by user of the DBDC system was lower than that of our system. This was attributable to the DBDC system's utterances that increased the incident of dialogue breakdown. When the DBDC system uses such utterances, the user responds with formulaic responses, such as "What do you mean?". Since the DBDC system frequently caused dialogue breakdowns, users used formulaic replies, and as a result, the number of vocabularies decreased.

6 Conclusions

In this study, we proposed a new utterance selection method called the NUR model for conversational dialogue systems. Our model ranks candidate utterances by their suitability in given contexts using neural networks. Our proposed model encodes utterances in context and candidates into fixed-length vectors, then processes these encoded vectors in chronological order to rank utterances. Experimental results showed that our proposed model ranked utterances more accurately than that of deep learning and other existing methods. In addition, we constructed a conversational dialogue system based on our proposed method and conducted experiments to evaluate its performance via dialogue with human subjects. By comparing the dialogue system of DBDC, we found our system able to conduct conversations more naturally than DBDC.

The dialogue system used in the experiment acquired topic words from given context in a simple manner. Because of this, there are some cases that the system selects inappropriate topics and fails in changing topics. Thus, future work includes topic management. Moreover, the system is unskilled at answering questions, and it often provokes dialogue breakdown. It requires a question-answering method corresponding to conversational dialogue systems.

Acknowledgements

This study received a grant of JSPS Grants-in-aid for Scientific Research 16H05880 and a research grant from Kayamori Foundation of Informational Science Advancement.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *Proc. ICLR*.

Rafael E Banchs and Haizhou Li. 2012. Iris: a chat-oriented dialogue system based on the vector space model. In *Proceedings of the ACL 2012*, pages 37–42. Association for Computational Linguistics.

[f] calculated using 100 dialogues

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Z. Cao, T. Qin, T.Y. Liu, M.F. Tsai, and H. Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

K.M. Colby. 1981. Modeling a paranoid mind. *Behavioral and Brain Sciences*, 4(4):515–560.

Antonella De Angeli and Rollo Carpenter. 2005. Stupid computer! abuse and social identities. In *Proceedings of the INTERACT 2005 workshop Abuse: The darker side of Human-Computer Interaction*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Ryuichiro Higashinaka, Nozomi Kobayashi, Toru Hirano, Chiaki Miyazaki, Toyomi Meguro, Toshiro Makino, and Yoshihiro Matsuo. 2014. Syntactic filtering and content-based retrieval of twitter sentences for the generation of system utterances in dialogue systems. *Proc. IWSDS*, pages 113–123.

Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. 2015a. Towards taxonomy of errors in chat-oriented dialogue systems. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 87–95.

Ryuichiro Higashinaka, Toyomi Meguro, Hiroaki Sugiyama, Toshiro Makino, and Yoshihiro Matsuo. 2015b. On the difficulty of improving hand-crafted rules in chat-oriented dialogue systems. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1014–1018.

Ryuichiro Higashinaka, Kotaro Funakoshi, Kobayashi Yuka, and Michimasa Inaba. 2016. The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics. In *10th edition of the Language Resources and Evaluation Conference*.

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

Michimasa Inaba, Sayaka Kamizono, and Kenichi Takahashi. 2014. Candidate utterance acquisition method for non-task-oriented dialogue systems from twitter. *Transactions of the Japanese Society for Artificial Intelligence*, 29(1):21–31.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. *Proc. EMNLP*, 3(39):413.

Makoto Koshinda, Michimasa Inaba, and Kenichi Takahashi. 2015. Machine-learned ranking based non-task-oriented dialogue agent using twitter data. In *2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, volume 3, pages 5–8.

Jen-Wei Kuo, Pu-Jen Cheng, and Hsin-Min Wang. 2009. Learning to rank from bayesian decision inference. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 827–836.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. *Proceedings of the NAACL-HLT 2016*.

R.D. Luce. 1959. *Individual choice behavior: A theoretical analysis*. Wiley, New York.

Colin L Mallows. 1957. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Hiroya Murao, Nobuo Kawaguchi, Shigeki Matsubara, Yukiko Yamaguchi, and Yasuyoshi Inagaki. 2003. Example-based spoken dialogue system using woz system log. In *4th SIGdial Workshop on Discourse and Dialogue*, pages 140–148.

Lasguido Nio, Sakriani Sakti, Graham Neubig, Toda Tomoki, and Satoshi Nakamura. 2014. Utilizing human-to-human conversation examples for a multi domain chat-oriented dialog system. *IEICE TRANSACTIONS on Information and Systems*, 97(6):1497–1505.

Kanako Onishi and Takeshi Yoshimura. 2014. Casual conversation technology achieving natural dialog with computers. *NTT DOCOMO Technical Jouranl*, 15(4):16–21.

RL Plackett. 1975. The analysis of permutations. *Applied Statistics*, pages 193–202.

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.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural Responding Machine for Short Text Conversation. *Proceedings of the 53th Annual Meeting of Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1577–1586.

Masahiro Shibata, Tomomi Nishiguchi, and Yoichi Tomiura. 2009. Dialog system for open-ended conversation using web documents. *Informatica*, 33(3).

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 NAACL-HLT 2015*, pages 196–205.

Hiroaki Sugiyama, Toyomi Meguro, Ryuichiro Higashinaka, and Yasuhiro Minami. 2013. Open-domain utterance generation for conversational dialogue systems using web-scale dependency structures. In *Proc. SIGDIAL*, pages 334–338.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proceedings of the ICML Deep Learning Workshop*, pages 1–7.

R.S. Wallace. 2008. The anatomy of A.L.I.C.E. *Parsing the Turing Test*, pages 181–210.

J. Weizenbaum. 1966. ELIZA-a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

A Dialogue Example

Table 5 shows example dialogue between our NUR-based dialogue system and a user. In the table, NB, PB, and B indicate respective rates of each label by 34 annotators.

Table 5: Examples of dialogues by our system (translated by authors)

Speaker	NB	PB	B	Utterance
System	-	-	-	こんにちは。最近はどんな感じ？ Hello. How have you been recently?
User	-	-	-	鼻炎がひどくてたいへんだよ。 My sinus is really bothering me.
System	100.0%	0.0%	0.0%	鼻炎の人って辛すぎるよ Sinus victims feel terrible.
User	-	-	-	そうなんだよ、くしゃみと鼻水が止まらなんだ。 That’s right. I can’t stop sneezing and have a runny nose.
System	100.0%	0.0%	0.0%	くしゃみする時の顔ってやばすぎるよ A sneezing face looks so terrible.
User	-	-	-	なので、マスクは必須アイテムです。 So a mask is an indispensable item
System	85.3%	14.7%	0.0%	くしゃみの音が大きいのが悩みだよ My loud sound of sneezing is more worrying.
User	-	-	-	たしかに、マスクじゃ音は止められないね。 I agree that the sound can’t be stopped by a mask.
System	94.1%	5.9%	0.0%	くしゃみやばいのは結構辛い Sneezing is bad for my health
User	-	-	-	鼻炎の薬飲んでみたらどうですか？ Why don’t you try sinus medicine?
System	58.8%	35.3%	5.9%	鼻炎が酷すぎて辛い My sinus is too terrible.
User	-	-	-	そうだよね、ティッシュペーパーがいくらあっても足りないよね。 Sure. You never have enough tissue paper, don’t you?
System	61.8%	32.4%	5.9%	お腹に不安がありすぎて迂闊に咳やくしゃみも出来ねえ Because of my concerns about my stomach, I can’t cough and sneeze.
User	-	-	-	やっぱり、医者で鼻炎のおクスリを貰ったらどうですか？ Ideally, why don’t you get medicine from a hospital?
System	58.8%	23.5%	17.6%	風邪っぽいのでクスリ飲むよ I seem to have a cold, so I’ll take medicine.
User	-	-	-	そりゃあ、早く飲んで栄養つけて休んだ方がいいよ。 You should take that, get nutrition and get some rest.
System	73.5%	26.5%	0.0%	お腹も弱いんだよね I also have a weak stomach.
User	-	-	-	じゃあ、おかゆでも食べる？ Then would you have rice gruel?
System	91.2%	8.8%	0.0%	味がついたおかゆはおいしい Flavored rice gruel is delicious.
User	-	-	-	はい、たまごを混ぜてお塩で味付けした粥です。 Yes, this rice gruel is flavored with egg and salt.
System	20.6%	44.1%	35.3%	照り焼きチキンとたまごのサンドイッチはおいしい Chicken teriyaki and egg sandwich is delicious.