

# Quote Recommendation in Dialogue using Deep Neural Network

Hanbit Lee\*, Yeonchan Ahn\*, Haejun Lee\*\*, Seungdo Ha\*, Sang-goo Lee\*

\*School of Computer Science and Engineering, Seoul National University, Seoul, Korea

\*\*Artificial Intelligence Team, Samsung Electronics, Seoul, Korea

\*{skcheon, acha21, seungtto, sglee}@europa.snu.ac.kr, \*\*haejun82.lee@samsung.com

## ABSTRACT

Quotes, or quotations, are well known phrases or sentences that we use for various purposes such as emphasis, elaboration, and humor. In this paper, we introduce a task of recommending quotes which are suitable for given dialogue context and we present a deep learning recommender system which combines recurrent neural network and convolutional neural network in order to learn semantic representation of each utterance and construct a sequence model for the dialog thread. We collected a large set of twitter dialogues with quote occurrences in order to evaluate proposed recommender system. Experimental results show that our approach outperforms not only the other state-of-the-art algorithms in quote recommendation task, but also other neural network based methods built for similar tasks.

## Categories and Subject Descriptors

I.2.7.7 [Artificial Intelligence]: Natural Language Processing—Text Analysis

## Keywords

Quote recommendation, Dialogue model, Deep neural network

## 1. INTRODUCTION

Quotes, or quotations, are well known phrases or sentences that we use in our writings and conversations for various purposes such as emphasis, elaboration, and humor. However, it is not always easy to promptly come up with an adequate quote that fits the situation at hand. We can try to find one on search engines, but it is also hard to figure out the right search keywords because in most cases the quotes are metaphorical so the words in those quotes do not always match the words expressing our intentions. A quote recommender can be quite useful in these situations.

Meanwhile, as we rely more and more on mobile devices in written communications including text messages, tweets,

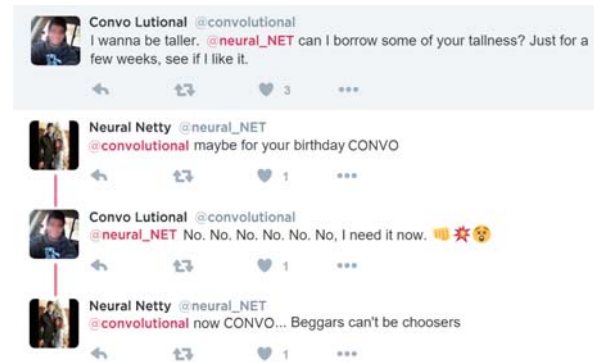


Figure 1: The example of tweet thread that includes the quote, "Beggars can't be choosers."

emails and blogs, convenience in text entry has become a very important feature for these devices. It is more so for wearable devices like Apple Watch and Samsung Galaxy Gear where the input/output spaces are constrained even further. An effective quote recommender, after monitoring an ongoing discourse, could recommend a few useful quotations from which the user can select instead of having to (laboriously) type in the entire sentence. Moreover, the system can be extended to recommend not only quotes but also other frequently-used responses as well.

Quote recommendation for writing (composition) has been introduced in Tan, et al., [10]. In this paper, we extend the task to dialogs. Dialogs differ from plain composition in that they consist of sequences of utterances, each of which forms a semantic unit. Thus, it is important to construct a recommendation model that is capable of capturing such sequential structure of dialogs.

We propose a deep neural network based quote recommendation system, which combines recurrent neural network (RNN) and convolutional neural network (ConvNet) in order to learn semantic representation of each utterance and construct a sequence model for the dialog thread.

As a dialog database, we use twitter dialog threads (threads where only two users are exchanging tweets). A twitter thread, or tweet thread, is a sequence of linked tweets, composed of an initial tweet and subsequent reply tweets. Figure 1 shows an example of a tweet thread that contains the quote, "Beggars can't be choosers". We use a large set of such threads with quote occurrences as the training set for the recommendation task.

Experimental results show that our approach outperforms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGIR '16, July 17-21, 2016, Pisa, Italy

© 2016 ACM. ISBN 978-1-4503-4069-4/16/07...\$15.00

DOI: <http://dx.doi.org/10.1145/2911451.2914734>

not only the other state-of-the-art algorithms for quote recommendation task, but also other neural network based methods built for similar tasks.

## 2. RELATED WORK

Quote recommendation can be viewed as a task of searching quote that fits in the given query texts. Tan et al., 2015 [10] apply learning-to-rank framework using cosine similarity-based features between query and quote context to recommend ranked list of quotes relevant to query. He et al., 2010 [2] also exploit similarity features between candidate paper and current writing context. Huang et al., 2012 [4] proposes to use translation model to connect the context of writing and the cited paper. These approaches use the traditional bag-of-words model which have a clear drawback: the loss of semantics. The bag-of-words model performs especially worse when it comes to dialogues because utterances in dialogues are in short and irregular colloquial forms.

With recent success of distributional word representation which overcomes such drawback, there are several approaches trying to learn distributional representation of sentences or paragraphs. One of the most prominent and recently emerging method is convolutional neural network. [5] Convolutional network based approaches are resulting in state-of-the-art performance for the tasks like sentence classification [6] and relevant short text matching [9]. We also take advantages of convolutional network to learn intermediate representation of each tweet.

Our work is related to dialogue models which have been studied for the purpose of building automatic dialogue systems. [13, 1] Recently with the significant advances in training of neural network, deep neural networks based dialogue models are actively being proposed. Most of them take recurrent neural network based approaches to model the utterance sequence. [12, 8] Their purpose is to generate natural language sentences, but our task focuses on recommending a ranked list of quotes among candidate quotes.

## 3. OUR DEEP NEURAL NETWORK MODEL

In this section we describe our deep neural network model for quote recommendation. Our architecture is a combination of the convolutional neural network (*ConvNetwork*) and the recurrent neural network (*RNN*) as shown in Figure 2. *ConvNetwork* maps tweets in the thread to their distributional vectors. And then the sequence of tweet distributional vectors are fed to *RNN* so as to compute the relevance of target quotes to the given tweet dialogue. In the following, we briefly explain main components of our network.

### 3.1 Recurrent Neural Network for tweet sequence modeling

Since use of a quote in a dialogue highly depends on temporal history of previous utterances, it is important to model a dialogue as a sequence of utterances. Recurrent neural network is a neural network which is widely used in modeling sequence, because it effectively learns significant patterns of sequential data. So we exploit recurrent neural network to recommend quotes suitable for given context of dialogue.

#### 3.1.1 Long short-term memory unit

We use long short-term memory unit (LSTM) [3] which is a recurrent neural network with several gates which allow networks to learn long-term dependencies without loss

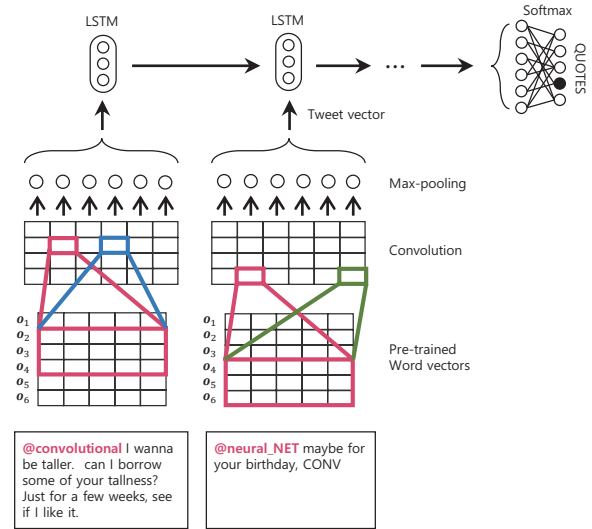


Figure 2: Overall architecture of our deep neural network model for quote recommendation

of information. LSTM retains latent vectors and updates those vectors by several gated functions with input vector for each time step. In our model we recurrently feed LSTM with distributional vector that represent each tweet in the tweet thread as shown in Figure 2. Formally, the input of LSTM is a tweet thread which is a sequence of tweet vectors:  $[v_1, v_2, \dots, v_n]$  where  $n$  is length of the tweet thread.

#### 3.1.2 Softmax

The output of LSTM layer is passed to fully connected layer with softmax activation in order to compute the probability of candidate quotes to be recommended. It computes probability of recommending quote  $j$  as follows:

$$p(y = j | x) = \frac{e^{x \cdot w_j}}{\sum_{k=1}^K e^{x \cdot w_k}}$$

where  $x$  is the output vector of LSTM,  $w_j$  is the weight vector of class  $j$  in fully-connected networks, and  $K$  is the number of class, in this case, the number of target quotes.

### 3.2 Convolutional Neural Networks for mapping tweets to intermediate vectors

We use convolutional neural networks to map tweets to their intermediate vectors. Convolution operation captures the significant local semantics, i.e., n-grams, of a tweet, so convolutional neural network can learn the optimal tweet representation for our quote recommendation task. Our *ConvNetwork* is composed of a single convolution layer and a max pooling layer.

#### 3.2.1 Convolution

The input to *ConvNetwork* is a tweet matrix  $t \in \mathbb{R}^{d \times m}$  where  $m$  is the number of words in the tweet and  $d$  is the dimensionality of word vectors.  $i$ -th column of  $t$  is  $d$ -dimensional word vector  $o_i \in \mathbb{R}^d$  corresponding to the  $i$ -th word in the tweet. Convolution layer captures the local patterns in tweet by convolving a filter matrix of weights  $w \in \mathbb{R}^{d \times h}$  with the input tweet matrix  $t \in \mathbb{R}^{d \times m}$ , where  $h$  is the filter width which is a hyper parameter. A feature

map  $\mathbf{c} \in \mathbb{R}^{m-h+1}$  is produced by convolution where each component is computed as follows:

$$c_i = f\left(\sum_{k,j} \mathbf{w} \circ \mathbf{t}_{[:,i:i+h-1]} + b_i\right)$$

where  $\circ$  is element-wise multiplication,  $\mathbf{t}_{[:,i:i+h-1]}$  is matrix slice of size  $h$ ,  $b_i$  is the bias value, and  $f$  is non-linear activation function. Generally, more than one filter is applied to convolution layer to yield richer information, so we use multiple filters  $\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \dots, \mathbf{w}^{(p)}$  where  $p$  is another hyper parameter which indicates the number of filters. With multiple filters, we get multiple feature maps  $\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \dots, \mathbf{c}^{(p)}$ .

### 3.2.2 Max-pooling

After obtaining feature maps, we apply max-pooling operation which take the maximum value  $\hat{c}^{(i)}$  from each of feature map  $\mathbf{c}^{(i)}$  and form a  $p$ -length feature vector:

$$\mathbf{v} = \langle \hat{c}^{(1)}, \hat{c}^{(2)}, \dots, \hat{c}^{(p)} \rangle$$

Eventually  $m$  tweet matrices in tweet thread are mapped to sequence of  $m$  tweet representation vectors.

$$[\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_m] \xrightarrow{\text{ConvNetwork}} [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m]$$

## 3.3 Training of the networks

We initialize the word vectors in our network with those trained from unsupervised neural language model and keep them static. It is a popular method to improve performance in case supervised training data is not sufficient. We use publicly available pre-trained word vectors, which were trained on 27 billion words of 2 billion tweets from Twitter. [7]

The entire networks are trained to minimize the cross-entropy of the predicted and true distributions. The objective function includes an  $L_2$  regularization term over the parameters to prevent overfitting. RMSProp [11] is used to optimize stochastic gradient descent with a learning rate of  $10^{-3}$  and a decay rate of 0.9. Dropout is adapted on max-pool layer in *ConvNetwork* and fully-connected layer in *RNN*. We set other hyper-parameters as follows: 3 for filter width ( $h$ ), 500 for the number of feature maps ( $p$ ), and dropout rate as 0.5. And we use ReLU activation function.

## 4. EXPERIMENTS

### 4.1 Data Construction

We collected the quotes to be recommended from two sources of quotes: one is the Wikiquote<sup>1</sup> website, and the other is Oxford Concise Dictionary of Proverbs<sup>2</sup>. For each quote, we collected tweet dialogues that contain at least one occurrence of the quote.

From each tweet dialogue thread, we only use the tweets up to the quote occurrence as *context*. For the tweet that contains the quote, only the part from the first word to the last word before the quote is used as *context*.

We started out with top 100 quotes in order of quantity of relevant tweet threads and then expanded to top 200, 300, and 400. Table 1 shows the statistics of datasets.  $q$  is total number of quotes,  $N$  is total number of tweet threads,  $t$  is average number of collected tweet threads per quote,

<sup>1</sup><http://en.wikiquote.org>

<sup>2</sup>Oxford University Press, 1998

**Table 1: Statistics of our dataset**

| Dataset | $q$ | $N$     | $t$    | $n$  | $m$  |
|---------|-----|---------|--------|------|------|
| ALL     | 412 | 222,645 | 540.4  | 3.85 | 13.9 |
| Top100  | 100 | 159,302 | 1593.0 | 3.94 | 13.4 |
| Top200  | 200 | 202,920 | 1014.6 | 3.90 | 13.6 |
| Top300  | 300 | 218,132 | 727.1  | 3.89 | 13.7 |
| Top400  | 400 | 222,408 | 556.0  | 3.89 | 13.7 |

$n$  is average number of tweets in a tweet thread, and  $m$  is average number of words in a tweet. We use 10% of dataset as validation set, another 10% as test set and rest of dataset as training set.

Originally there is only one gold quote for each query context in test data, but there are many cases in which more than one quote can be used. For example, the quote “Great minds think alike.” can be substituted with the quote “Birds of a feather flock together.” in the situation when two people share an idea. We sampled test queries from the top100 dataset and manually tagged the appropriate quotes that can be used in each query context. (*MultipleGold*)

## 4.2 Baselines and Evaluation Metrics

### 4.2.1 Baselines

We compare our approach with three state-of-the-art content based recommendation approaches [2, 4, 10]. Context-aware relevance model (CRM) recommends a quote based on similarity between the query and the contexts of the quote using bag-of-words representation. Citation translation model (CTM) applies a translation model to quote recommendation by using the probability that the query context would be translated into the quote. Learning-to-recommend quote (LRQ) is most recent approaches that use learning-to-rank framework with quote-based features, quote-context similarity features, and context similarity features. We also experiment with CNN and RNN separately to compare with our combined networks. Since these simple neural models cannot accept tweet sequence as input, so we concatenate all tweets in a tweet thread into one long context tweet and use it as the input for the particular thread.

### 4.2.2 Evaluation Metrics

**MRR:** Mean reciprocal rank (MRR) is the average score of inverse rank of the gold quote in recommendation list.

$$MRR = \frac{1}{|test|} \sum \frac{1}{rank(gold_{quote})}$$

**Recall@k:** Since there is only one gold quote for each query in original test set, Recall@k is the number of cases that the gold quote is recommended in top-k result divided by total test cases.

**NDCG@k:** Normalized discounted cumulative gain (NDCG) is a widely used evaluation metric in various IR tasks. It takes into account the ranking position of gold quotes in recommendation list.

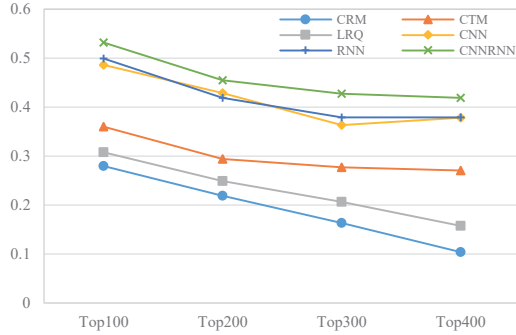
**Hit@k:** For MultipleGold test set, we use Hit@k which is the number of cases that any of gold quotes are recommended in top-k result divided by total test cases.

## 4.3 Results and Discussion

The comparison results of different methods for the top100 dataset are shown in table 2. As we can see, CRM and LRQ

**Table 2: Recommendation performance of different methods for top100 dataset**

| Method | MRR          | Recall@5     | NDCG@5       | Hit@5        |
|--------|--------------|--------------|--------------|--------------|
| CRM    | 0.207        | 0.279        | 0.198        | 0.470        |
| CTM    | 0.274        | 0.359        | 0.269        | 0.579        |
| LRQ    | 0.221        | 0.308        | 0.214        | 0.523        |
| CNN    | 0.378        | 0.486        | 0.385        | 0.668        |
| RNN    | 0.375        | 0.489        | 0.389        | 0.661        |
| CNNRNN | <b>0.416</b> | <b>0.532</b> | <b>0.422</b> | <b>0.698</b> |



**Figure 3: Results of recall@5 of different methods when increasing # of target quotes**

which search for quotes based on similarity features between query and quote contexts, do not result in successful performance. That is because they cannot find out contexts which are semantically similar to the query but expressed in different words. CTM which use translation model performs slightly better. Deep neural network based approaches achieve higher scores than the non-neural methods, which shows that the distributed word representation helps semantic word matching and the deep neural network automatically extract meaningful representation of tweet context in respect to the quote recommendation. CNN captures significant local semantic features, i.e., n-grams, while RNN captures the overall ordering of words in context. Our approach outperforms both separate CNN and RNN models showing that per-tweet feature extraction captures structural context of dialogue while simple concatenation of tweets cannot. Figure 3 shows the results of recall@5 when increasing the number of target quotes. Although overall accuracy decreases as the number of target quotes increases, our approach consistently outperforms other methods.

Hit@5 evaluation of our model reaches almost 0.7 for top100 dataset which means that our recommendation system recommends at least one relevant quote in top 5 list in seven out of ten times. This proves that our recommendation model is practically useful. Figure 4 shows the example of recommendation results for a query context in the test set. As we can see, our system recommends quotes comforting the user named “convolution” and it is quite appropriate to the situation.

## 5. CONCLUSION

In this work, we present a task of quote recommendation in a dialogue. We propose a deep neural network model which efficiently extracts meaningful local semantic features from each tweet using convolutional network and learns each tweet features in sequence. We evaluate the proposed model

| Query context  | Results                               |
|--|---------------------------------------|
| @convolution Ahh! I’m so excited now looking forward to September!                   | 1.Slow and steady wins the race       |
| @neural_NET yay me too!!! I didn’t think u would still be in Orlando in September!!! | 2.Business before pleasure            |
| @convolution yea...things are taking longer than expected                            | 3.Patience is a virtue                |
| @neural_NET hey don’t worry!   | 4.All good things must come to an end |
| [Quote]  | 5.Better safe than sorry              |

**Figure 4: An example results of quote recommendation**

on real twitter dialogue data to show that our approach is effective for quote recommendation. In the future, we plan to investigate several useful dialogue features that may not be captured via neural networks.

## 6. ACKNOWLEDGEMENT

This paper was supported by Samsung Electronics.

## 7. REFERENCES

- [1] M. Gasic, C. Breslin, M. Henderson, D. Kim, M. Szummer, B. Thomson, P. Tsiakoulis, and S. Young. On-line policy optimisation of bayesian spoken dialogue systems via human interaction. In *ICASSP*, 2013.
- [2] Q. He, J. Pei, D. Kifer, P. Mitra, and C. L. Giles. Context-aware citation recommendation. In *WWW*, 2010.
- [3] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [4] W. Huang, S. Kataria, C. Caragea, P. Mitra, C. L. Giles, and L. Rokach. Recommending citations: Translating papers into references. In *CIKM*, 2012.
- [5] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *ACL*, 2014.
- [6] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.
- [7] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.
- [8] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *ICASSP*, 2013.
- [9] A. Severyn and A. Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *SIGIR*, 2015.
- [10] J. Tan, X. Wan, and J. Xiao. Learning to recommend quotes for writing. In *AAAI*, 2015.
- [11] T. Tieleman and G. Hinton. Lecture 6.5 - rmsprop, coursera: Neural networks for machine learning. In *Technical report*, 2012.
- [12] O. Vinyals and Q. V. Le. A neural conversational model. In *arXiv*, 2015.
- [13] S. Young, M. Gasic, B. Thomson, and J. D. Williams. Pomdp-based statistical spoken dialog systems: A review. *IEEE*, 101(5):1160–1179, 2013.