Recursive Neural Network Paraphrase Identification for Example-based Dialog Retrieval

Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, Satoshi Nakamura
Graduate School of Information Science, Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
E-mail: {lasguido.kp9,ssakti,neubig,tomoki,s-nakamura}@is.naist.jp

***Abstract*—An example-based dialog model often require a lot of data collections to achieve a good performance. However, when it comes on handling an out of vocabulary (OOV) database queries, this approach resulting in weakness and inadequate handling of interactions between words in the sentence. In this work, we try to overcome this problem by utilizing recursive neural network paraphrase identification to improve the robustness of example-based dialog response retrieval. We model our dialog-pair database and user input query with distributed word representations, and employ recursive autoencoders and dynamic pooling to determine whether two sentences with arbitrary length have the same meaning. The distributed representations have the potential to improve handling of OOV cases, and the recursive structure can reduce confusion in example matching.**

I. Introduction

Rule based dialog system is a popular technique in utilizing example conversation to generate a hand-made rule for chat-oriented dialog system response generation. This techniques achieved a current state of the art in the intelligent machine test so called the Turing Test[1]. To simplify work, rule based technique is often created with an open source tools [1] that easy to use. Behind it success, the rule based technique rely on the humans hand-made rule that is complicated and difficult to be expanded.

In other hand, the data-driven approach is not much explored technique that could overcoming this shortage. Given the user input, this method search and generate a response based on the conversation database. Comparing to the rule based technique [2], [3], data-driven based doesn't rely on the complicated hand-made rules and easy to expanded. This approach allows for the use of large amounts of data on the Web to efficiently find responses for a large variety of user queries, and becomes popular as lightweight methods to create broad-coverage chat-oriented dialog systems [4], [5], [6], [7]. However, To achieve the good coverage, recording of a large data set of real human-to-human conversation is necessary, and some studies propose constructing dialog examples from available log databases created using Wizard of OZ (WOZ) systems [8] or Twitter [9].

An example based dialog modeling (EBDM) is one of many approach to data-driven framework. It works by matching the user's utterance with a query in the query-response database, then returning the response that corresponds with the most closely matching query. By retrieving examples from a database and displaying the response to the user, EBDM is only able to generate examples that are actually included in the database. Because of this, it is able to generate highly natural output when a response is included in the database and the example is able to be appropriately retrieved [4], [5], [6]. However, we can think of a number of situations in which these simplistic methods are clearly inadequate. For example, if the user's utterance is "it is not raining today," previously proposed matching methods will give "it is raining today" a high score, and the system may provide the exact opposite response a user desires. When the system is not able to find similar examples to determine the response, most EBDM systems currently rely on either canned or template response which may result in less than satisfactory output [10], [11].

In general, two factors contribute greatly to the accuracy of EBDM systems: the coverage of the dialogue corpus, and the effectiveness of the example retrieval. We focus particularly on the latter of these problems in EBDM systems, arguing that more sophisticated methods for matching user utterances and queries in the database are necessary. We note that compositional distributional representation using neural networks [12], [13] may have a potential to capture a large number of linguistic phenomenon or a simpler one such as paraphrase. In this work, we introduce a new approach in applying these representations to matching the user utterance and queries in the database. Furthermore, we utilized dialog corpora constructed from movie conversation data as a testbed for the proposed method.

II. Related Work

There have been a number of related work in response retrieval and paraphrase detection field. It utilizes several feature raging from the lexical matching feature [14], [15], [16], Wordnet-based semantic similarity measure [17], [18], and syntactic measure [19].

A method to detect paraphrase with a compositional distributional representations using neural networks previously proposed by Socher et al. [20]. This method had become a state of the art of paraphrase detection method. Using a similar approach, we build a paraphrase detection model out of dialog pair conversation corpus. Later, we utilize our paraphrase model to retrieve an appropriate response in the

[1] http://people.exeter.ac.uk/km314/loebner2013/index.php

978-616-361-823-8 © 2014 APSIPA

dialog database to the user, especially when the system can not find any match response in the example database (OOV case).

III. Overall Dialog System

Figure 1 depicts an overview of our dialog system. User input is treated by the dialog management system as a query to our response generator module. Given the user input, our response generator will search appropriate response through the database with the EBDM response retrieval in the first place. If there is no matched or related example dialog pair in the database, response generator will encounters OOV problem. At this point, usually the system responses will out of topics and not related to the user input. We overcome this condition by using the paraphrase retrieval to retrieve the appropriate response from the example database. To be noted that, in this paper we focus on exploring the neural-network-based retrieval performance.

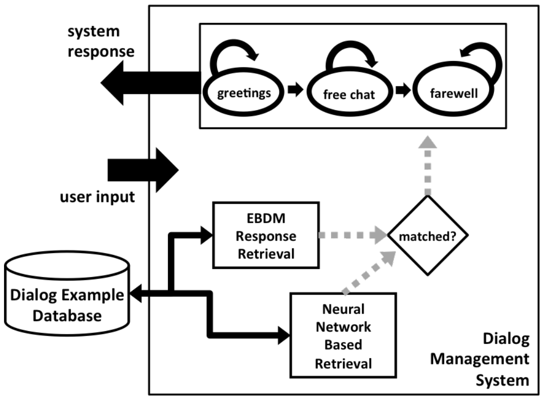


Fig. 1. Dialog Management System.

IV. Neural Network Based Retrieval

A proper system response is retrieved by modeling example database into neural word representation and calculating the probability between the user input and an example in the database. An overview of the paraphrased based retrieval is depicted in Figure 2.

Adopting the work of [20], we utilize recursive autoencoders (RAE), dynamic pooling, and a softmax classifier to decide weather the sentence is paraphrased or not. In the following section we describe about: (1) neural language model, which computes a word representation as an input to the RAE, (2) recursive autoencoders, and (3) dynamic pooling.

A. Word Representations

A distributed word representation is a n-dimensional vector of continuous values used to represent a word in the vocabulary. They are often obtained by joint learning of neural network language models and distributed representation for words [21]. Improved word representations [13] are known to capture distributional syntactic and semantic information via the word co-occurrence statistics. In a word representation, each word in dictionary ($i \in D$) is embedded into n-dimensional space $L \in \mathbb{R}^{n \times |D|}$. From this representation, a word vector can be seen as a single vector in the column L.

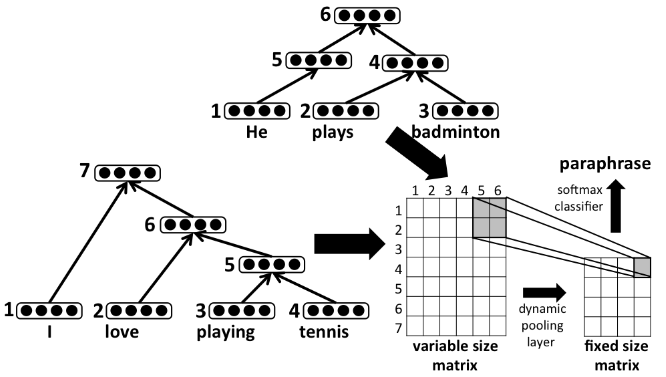


Fig. 2. Overview of neural-network-based retrieval.

B. Recursive Autoencoder

By using the RAE algorithm, we combine word representations in a syntactic parse tree into a vector representations of longer phrases. This approach is intends to capture the compositionality of meaning that is naturally constrained by the syntactic parse tree. To construct the final vector representation, this algorithm takes a word representations and a binary syntactic tree as an input.

In this algorithm, every child and non-terminal node in the binary tree is collected as a feature representation of a sentence. The binary tree forms the parent and children triplets ($p \rightarrow c_1 c_2$) where each child could be a word representations vector or the other non-terminal nodes. A parent p is calculated through the neural network layer (Equation (1))

$$p = f(W_e[c_1; c_2] + b), \tag{1}$$

where $[c_1; c_2]$ is concatenation of the two children and f is a tanh activation function. The weight (W_e) and bias (b) value are trained using recursive autoencoders [20].

C. Dynamic Pooling

After obtaining the RAE-derived representation of the sentence, next we would like to calculate the similarity of two sentences. In order to do so, we need to deal with the arbitrary length of the sentence. Thus, we need to normalized the RAE word representations into a fixed length vector with an algorithm called dynamic pooling. As we described previously, every sentence fed into the RAE forms a binary tree representation. Given this, we can define a matrix M, where the rows and columns in this matrix represent two sentences with the different lengths i and j. This row and column represents the non-terminal nodes and leaves in the binary tree, therefore the matrix M's size is $2i - 1 \times 2j - 1$.

This algorithm takes a matrix M as an input and output a matrix M' with the fixed size $n \times n$. In this algorithm, matrix M will be divide into n roughly equal parts. Every minimal value in the rectangular window is selected to form a $n \times n$ grid. During this process, matrix M' will lose some part of the information compared to the original matrix M [20]. But this approach manages to capture the matrix M's global structure.

Given the uniform size of matrix M' from every sentence, we classify each utterance similarity using a softmax classifier layer afterwards. The softmax classifier takes the matrix M' as an input, and outputs a confidence score that decided whether a user input and dialog database is a paraphrase or not.

V. Experimental Setup

A. Dialog Corpora

In this work, we use a movie scripts to build our dialog corpus. We collect our movie script dialogues based on Friends TV show scripts[2], The Internet Movie Script Database[3], and The Daily Script[4]. The total number of gathered movie scripts is 1,786 with 1,042,288 dialog pairs. More details on the data can be found in [22]

In our movie dialog corpus, we define two basic types of information for each dialog: actor and utterances. The utterances are the actual content of each dialog turn in the movie scripts. The actor refers to the character name in the movies. This actor and utterance information will be utilized in construct the dialog corpus.

B. Filtering

In our previous study, we have introduce a *tri-turn* unit to find the candidate of dialog-pairs [23]. Given the HTML format of movie script as input, we construct a dialog corpus by perform a filtering as follows,

1 Preprocessing, which removes unnecessary information and normalize the text. This step is done by transforming raw HTML files into raw text format. To transform a variety of sources of movie scripts that had various formats, we implemented several parsing algorithms to fetch the information from the raw movie conversation. We also remove unnecessary explanatory information about the movie scenes.

2 The extraction of a dialog-pairs, which ensures that the conversation is done between two people talking each other. We perform dialog (*tri-turn*) extraction to find the candidate of dialog-pairs to construct appropriate dialog-pair examples from raw movie scripts files. A tri-turn is a three conversation turns between two actors X and Y that has the pattern X-Y-X. In one tri-turn, the first and last dialog turn are performed by the same actor and the second dialog turn is performed by the other actor. Later the query-response pairs are made by separating the tri-turn pattern X-Y-X into two pairs, X-Y and Y-X.

3 Semantic similarity calculation, which ensure that the each query-response pair is semantically related. To ensure the semantic relationship between dialog-pairs, semantic similarity [24] as shown in Equation (2) is performed. The similarity of sentence X and Y can be obtained by calculating the relation between X_{syn} and Y_{syn}. X_{syn} and Y_{syn} respectively is a group of WordNet[5] synsets for each word in the sentence X and Y that are linked by a complex network of lexical relations. Every dialog pair with high similarity is included as an example database.

$$sem_{sim}(X,Y) = \frac{2 \times |X_{syn} \cap Y_{syn}|}{|X_{syn}| + |Y_{syn}|} \tag{2}$$

C. NN-based Retrieval Setup

In our experiment, we use the trained RAE with 150,000 sentences from NYT and AP section of the Gigaword corpus provided by Socher et al. [20]. To generate all the parse trees for the RAE algorithm, we use the Stanford parser [25]. We also employ the 100-dimensional word representations computed and provided by Turian et al. [26]. Furthermore, we use natural language processing tools and Wordnet synsets provided by the NLTK toolkit[6], and Apache Lucene[7] to calculate the TF-IDF based cosine similarity.

After performing pre-processing, filtering, and picking some words to be transformed into a vector of word representations, we finally use 10,033 dialog pairs as our training and test data. During the experiment, we randomly separate our dialog pairs data into 1,000 and 9,033 pairs consecutively as test and train dialog pairs data.

When training the NN-based retrieval model, we need to consider that the NN-based retrieval model is also rely on the softmax classifier layer. This layer decide whether a user query and dialog database is similar or not. To provide a balance amount of similar and not similar query during training, we do cross product all the training dialog (9,033 pairs) with each other and calculate the syntactic-semantic similarity (see Equation 3) between them. We assume that the similar query is obtained when the syntactic-semantic score is exclusively between 0.7 and 0.9, and not similar query is obtained when the syntactic-semantic score is exclusively between 0.2 and 0.4. In the end, we got 1,421,338 pair of dialog train with the ratio between similar and not similar sentence is 50:50 .

$$sim(S_1,S_2) = \alpha[sem_{sim}(S_1,S_2)] + (1-\alpha)[cos_{sim}(S_1,S_2)] \tag{3}$$

VI. Performance of NN-Based Retrieval

Table I shows the correlation between user input and example database. We calculate syntactic-semantic score sim for each utterance pair ($S1$ and $S2$). We observed that when a utterance pair have a high similarity score (a similar pair), it will generates a clear diagonal structure of dark line in the matrix representation. This matrix shows the paraphrase relations between two utterances. A clear diagonal structure of dark line in the matrix was a result from the Euclidean distance computation. During this case, the NN-based retrieval is manage to find a close/paraphrased sentence to the input query.

[2] http://ufwebsite.tripod.com/scripts/scripts.htm

[3] http://imsdb.com/

[4] http://dailyscript.com/

[5] http://wordnet.princeton.edu/

[6] http://nltk.org

[7] http://lucene.apache.org/

In NN-based retrieval (RNN) approach, given the user input the system will find a paraphrased input sentence in the dialog pair example database. Later, it will outputs the corresponding response from the matched dialog pair. The comparison of the RNN with EBDM TF-IDF based cosine similarity (ebdm) can be seen in the table II.

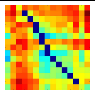
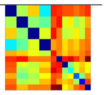
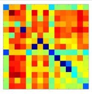
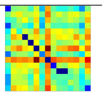
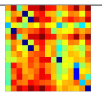
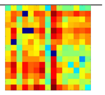
sim	Sentences	Matrix
0.94	S_1) Captain, we can not keep going fast on these icy roads. S_2) We can not keep going fast on these icy roads!	
0.93	S_1) I'll see you there. S_2) I'll see you. for say to myself?	
0.65	S_1) So what do you have to say for yourself? S_2) Why should I have anything for say to myself?	
0.60	S_1) Hold your fire! He's got a girl. S_2) Looks like he's got a hostage.	
0.50	S_1) I've been careful, I've been waiting my chance. S_2) Oh, you've been under a lot of stress.	
0.38	S_1) Yes, I can see that too and I don't think it's so terrible. S_2) That's why I do all the thinking.	

TABLE I
SENTENCE PAIRS.

user input	ebdm response	rnn response
I own this store.	Who told you to buy a brownstone on my block, in my neighborhood on my side of the street?	What are you talking about? You love your job.
I am not!	I can't speak for my friend, but my friend is in a loving, committed relationship.	You're wrong.
That look you just gave me?	But you gave me a ticket to the Super Bowl. Hand it over, my friend.	I do. I want to very badly.

TABLE II
EBDM AND RNN RESPONSE COMPARISON.

VII. Conclusion

In this paper, we investigated recursive neural network paraphrase identification techniques in order to retrieve responses in a data-driven chat-oriented dialog system. The experiment shows that the neural network based retrieval is able to capture the correlation between user input and example database especially when the user input is not available in the example database (OOV case). As future work, exploring and applying the neural network word representations into dialog example database can be a promising future direction.

Acknowledgment

Part of this work was supported by JSPS KAKENHI Grant Number 26870371.

References

[1] R. S. Wallace, "The anatomy of a.l.i.c.e." in *Parsing the Turing Test*, R. Epstein, G. Roberts, and G. Beber, Eds. Springer Netherlands, 2009, pp. 181–210.

[2] J. Weizenbaum, "Eliza - a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[3] R.Wallace, *Be Your Own Botmaster*. A.L.I.C.E A.I. Foundation, 2003.

[4] S. Jung, C. Lee, and G. Lee, "Dialog studio: An example based spoken dialog system development workbench," in *Proc. of the Dialogs on dialog: Multidisciplinary Evaluation of Advanced Speech-based Interactive Systems. Interspeech 2006-ICSLP satellite workshop,*, Pittsburgh, USA, 2006.

[5] C. Lee, S. Lee, S. Jung, K. Kim, D. Lee, and G. Lee, "Correlation-based query relaxation for example-based dialog modeling," in *Proc. of ASRU*, Merano, Italy, 2009, pp. 474–478.

[6] K. Kim, C. Lee, D. Lee, J. Choi, S. Jung, and G. Lee, "Modeling confirmations for example-based dialog management," in *Proc. of SLT*, Berkeley, California, USA, 2010, pp. 324–329.

[7] A. Ritter, C. Cherry, and W. B. Dolan, "Data-driven response generation in social media," in *Proc. of EMNLP*, Edinburgh, Scotland, UK., July 2011, pp. 583–593.

[8] H. Murao, N. Kawaguchi., S. Matsubara, Y. Yamaguchi, and Y. Inagaki, "Example-based spoken dialogue system using WOZ system log," in *Proc. of SIGDIAL*, Saporo, Japan, 2003, pp. 140–148.

[9] F. Bessho, T. Harada, and Y. Kuniyoshi, "Dialog system using real-time crowdsourcing and twitter large-scale corpus," in *Proc. of SIGDIAL*, Seoul, South Korea, 2012, pp. 227–231.

[10] C. Lee, S. Jung, S. Kim, and G. G. Lee, "Example-based dialog modeling for practical multi-domain dialog system," *Speech Commun.*, vol. 51, no. 5, pp. 466–484, May 2009.

[11] N. Chambers and J. Allen, "Stochastic language generation in a dialogue system: Toward a domain independent generator." in *Proc. of SIGDIAL*, Cambridge, Massachusetts, USA, 2004, pp. 9–18.

[12] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proc. of EMNLP*, 2012.

[13] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. of ICML*, New York, NY, USA, 2008, pp. 160–167.

[14] R. E. Banchs and H. Li, "IRIS: a chat-oriented dialogue system based on the vector space model," in *Proc. of ACL (System Demonstrations)*, 2012, pp. 37–42.

[15] R. E. Banchs, "Movie-dic: a movie dialogue corpus for research and development." in *Proc. of ACL*, 2012, pp. 203–207.

[16] L. Qiu, M.-Y. Kan, and T.-S. Chua, "Paraphrase recognition via dissimilarity significance classification," in *Proc. of EMNLP*, Stroudsburg, PA, USA, 2006, pp. 18–26.

[17] L. Nio, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Combination of example-based and smt-based approaches in a chat-oriented dialog system," in *Proc. of ICE-ID*, 2013.

[18] A. Islam and D. Inkpen, "Semantic similarity of short texts," in *Proc. of RANLP*, Borovets, Bulgaria, September 2007.

[19] D. Das and N. A. Smith, "Paraphrase identification as probabilistic quasi-synchronous recognition," in *Proc. of ACL-IJCNLP*, Stroudsburg, PA, USA, 2009, pp. 468–476.

[20] R. Socher, E. H. Huang, J. Pennington, A. Y. Ng, and C. D. Manning, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Advances in Neural Information Processing Systems 24*, 2011.

[21] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[22] L. Nio, S. Sakti, G. Neubig, T. Toda, and S. Nakamura, "Utilizing human-to-human conversation examples for a multi domain chat-oriented dialog system," *IEICE Transactions on Information and Systems*, June 2014.

[23] L. Nio, S. Sakti, G. Neubig, T. Toda, M. Adriani, and S. Nakamura, "Developing non-goal dialog system based on examples of drama television," in *Proc. of IWSDS*, Paris, France, 12 2012.

[24] D. Liu, Z. Liu, and Q. Dong, "A dependency grammar and wordnet based sentence similarity measure," *Journal of Computational Information Systems*, vol. 8, no. 3, pp. 1027–1035, 2012.

[25] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proc. of ACL*, Stroudsburg, PA, USA, 2003, pp. 423–430.

[26] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proc. of ACL*, 2010.