Active Learning for Example-based Dialog Systems

Takuya Hiraoka and Graham Neubig and Koichiro Yoshino and Tomoki Toda and Satoshi Nakamura

Abstract While example-based dialog is a popular option for the construction of dialog systems, creating example bases for a specific task or domain requires significant human effort. To reduce this human effort, in this paper, we propose an active learning framework to construct example-based dialog systems efficiently. Specifically, we propose two uncertainty sampling strategies for selecting inputs to present to human annotators who create system responses for the selected inputs. We compare performance of these proposed strategies with a random selection strategy in simulation-based evaluation on 6 different domains. Evaluation results show that the proposed strategies are good alternatives to random selection in domains where the complexity of system utterances is low.

1 Introduction

Example-based dialog is one popular method for constructing dialog systems [1]. Example-based dialog managers store dialog examples, which consist of pairs of an example input and a corresponding system response, in a database, then generate system responses for input based on these dialog examples. Example-based dialog managers can be easily and flexibly modified by updating dialog examples in the database, and thus are effective in situations where 1) the domain or task of the dialog system is frequently expanded, or 2) constructing a sophisticated dialog manager a priori is difficult. In previous research, this variety of dialog managers has been used for information retrieval dialog systems [1, 2, 3], multi-domain dia-

Takuya Hiraoka, Graham Neubig, Koichiro Yoshino, Satoshi Nakamura
Nara Institute of Science and Technology (NAIST) e-mail: {takuya-h,neubig,koichiro,s-nakamura}@is.naist.jp

Tomoki Toda
Nagoya University e-mail: tomoki@icts.nagoya-u.ac.jp

log systems [4], question answering dialog systems [5], and chatter-oriented dialog systems [6, 7, 8].

Generally, in the construction of example-based dialog managers, a large number of dialog examples are required to cover a variety of inputs in the dialog. To deal with this problem, Banchs et al. [6] and Nio et al. [8] utilize dialog corpora acquired from the Web (e.g. Twitter posts or movie scripts) as dialog examples. However, generally, corpora on the Web include examples which might have a bad influence on the dialog system's performance (e.g. ungrammatical or impolite sentences), and manual screening by a human is needed. In addition, we cannot always find dialog corpora that match the dialog system's domain and style. Therefore, manual creation of dialog examples is still required in the development of practical example-based dialog managers.

In this paper, we propose a method that reduces the human effort in creating dialog examples by using active learning [9] to construct an example-based dialog manager. Given 1) a prototype example-based dialog system with a small example base and 2) input logs of the prototype system, we focus on improving the example base in the prototype dialog manager by adding new dialog examples (pairs of an input and the corresponding system response) efficiently. At first, in Section 2, we propose an active learning framework for construction of example-based dialog managers that employs some strategy to determine which inputs should be labeled with system responses. In Section 3, a couple of strategies for selecting effective examples are proposed. In Section 4, we evaluate the proposed strategies with simulated active learning experiments.

The main contribution of this paper is that, to our knowledge, this is the first work that applies active learning to construction of a database (such as dialog examples in this research) for a dialog system. In the context of dialog research, active learning has mainly been applied to construction of language understanding [10, 11, 12, 13, 14, 15, 16] and speech recognition modules [17, 18]. Mairesse et al. [19] use active learning in construction of natural language generation module. Further, Gasic and Young [20] use active learning to speed reinforcement learning of the dialog system policy. Unlike these related works, we apply active learning to the construction of a database (i.e. example base) for a dialog system.

2 An active learning framework for example-based dialog managers

In this section, we describe example-based dialog managers, and the proposed active-learning framework.

2.1 Example-based dialog managers and their evaluation

Example-based dialog managers utilize dialog examples to respond to input. Dialog examples $D := \{\langle u_i, s_i \rangle\}_{i=1}^{|D|}$ consist of pairs of an example input u_i (e.g., a user utterance or a system dialog state) and a corresponding system response s_i (left side of Figure 1). Given the example base D, the dialog manager determines the system response s^* to input u by the following steps:

1. Calculate the similarity $\text{sim}(u_i, u)$ between all example inputs u_i in D, and input u. This is often defined as tf-idf weighed cosine similarity [21]:

$$\text{sim}(u_i, u) := \frac{\text{w}(u_i) \cdot \text{w}(u)}{|\text{ w}(u_i) \text{ }| \cdot |\text{ w}(u) \text{ }|} \tag{1}$$

   where the function w returns the vector representation of input (for example the frequency vector of the content words) weighted according to tf-idf.
2. Return system response s^* whose corresponding example input u^* has the highest similarity with u:

$$u^* = \arg\max_{u_i \in D} \text{sim}(u_i, u) \tag{2}$$

$$s^* = \{s_i | \langle u_i, s_i \rangle \in D \wedge u_i = u^*\} \tag{3}$$

The left side of Figure 1 demonstrates how the system determines a response for the user input “That’s fun!”, calculating the similarity between this input and example user inputs in D based on Eq. (1). The similarity between “Football is fun!” (u_{54}) and the user input is 0.6, which is the highest of the example inputs in D. Therefore, based on Eqs. (2) and (3), “Seems to be fun.” (s_{54}), which is the system utterance corresponding to example user input u_{54}, is selected as system response s^*. This method is commonly used in the core of example-based dialog managers [1, 2, 3, 4, 6, 8].

Given an example-based dialog manager, it is necessary to evaluate the quality of its responses. To maintain generality of our framework, we avoid using a domain specific evaluation framework (such as task-completion), and use reference-based evaluation [8, 7, 22, 23, 24] instead. In particular, we follow the evaluation framework of Nio et al. [8] and evaluate the dialog system with test examples (right side of Figure 1). The test examples $T = \{\langle u_m, s_m \rangle\}_{m=1}^{|T|}$ consist of pairs of a test input u_m and the oracle system response s_m. Using these test examples, we calculate average similarities between the dialog system’s responses and the oracle system responses for the evaluation. More concretely, given test examples T and the dialog system S, the performance p of S is calculated as follows:

$$p = \frac{1}{|T|} \sum_{m=0}^{|T|} \frac{\text{w}(s_m^*) \cdot \text{w}(s_m)}{|\text{ w}(s_m^*) \text{ }| \cdot |\text{ w}(s_m) \text{ }|}. \tag{4}$$

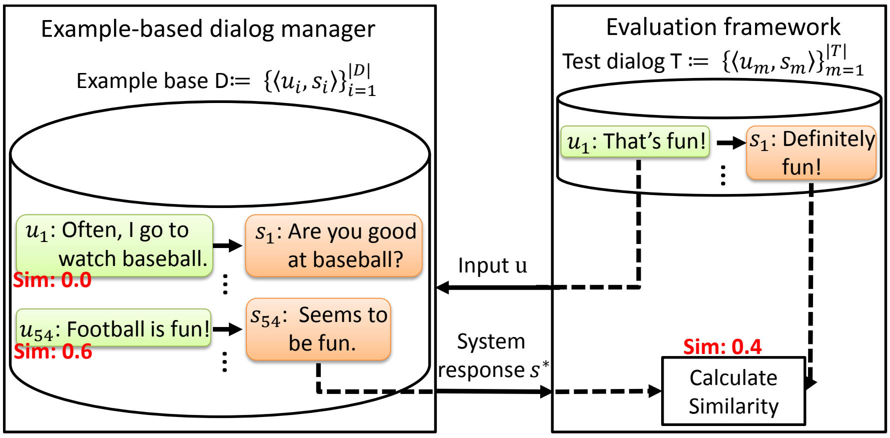

Fig. 1 Example-based dialog managers and their evaluation.

This is the average of cosine similarities between S's response s_m^*, calculated according to Eq. (1), and the oracle system response s_m over the test examples.[1] $|T|$ represents the total number of pairs in test set T. In the example in the right side of Figure 1, the evaluation framework evaluates the system response to the test user input "That's fun!" (u_1). In this example, the system outputs "Seems to be fun." as its response to u_1. The similarity between "Seems to be fun." and oracle system response "Definitely fun!" is calculated according to Eq. (4).

2.2 *Active learning framework*

In this section, we propose an active learning framework for the example-based dialog managers described in Section 2.1. Starting from a prototype example-based dialog manager with a small number of dialog examples, the active learning problem is to improve the system as much as possible with minimal human effort. We focus on the situation where there are input logs collected by the prototype dialog system, and a human creator is required to create system responses for these inputs (Figure 2). Therefore, given the example dialog $D := \{\langle u_i, s_i \rangle\}_{i=1}^{|D|}$ and input log $U := \{\langle u_j \rangle\}_{j=1}^{|U|}$, the goal is to select the *subset of input that yields the greatest improvement in system performance* from U to present to the human creator.

Algorithm 1 describes our active learning framework in detail. At first, we construct our initial system S with example base D, and evaluate its performance based on test data T using Eq. (4) (line 2 and line 3). Then, we continue to incrementally update dialog examples for S until training epoch e reaches a particular threshold

[1] The experimental results of Nio et al. [8] indicate that the human subjective evaluation for naturalness and relevance of system response is correlated with the score calculated in Eq. (4).

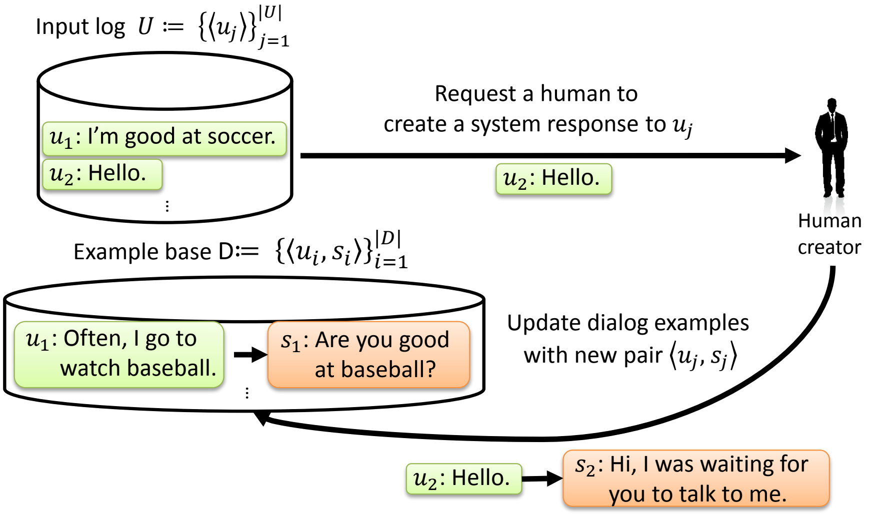

Fig. 2 Active learning for updating the example base given input logs.

Algorithm 1 AL-EBDM

1: **Input** example base D, input log U
2: $S \leftarrow$ **constructSystem**(D)
3: Evaluate the performance of S on the test data T
4: **for** e=1,2,... **do**
5:     Select k inputs $\{u_1, \ldots, u_k\}$ from U, and request a human creator to create system response $\{s_1, \ldots, s_k\}$.
6:     Remove $\{u_1, \ldots, u_k\}$ from U.
7:     $D \leftarrow D \cup \{\langle u_1, s_1 \rangle, \ldots, \langle u_k, s_k \rangle\}$
8:     $S \leftarrow$ **constructSystem**(D)
9:     Evaluate the performance of S on the test dialog T
10: **end for**

(from line 4 to line 10). Lines 5 and 6 select and remove k inputs from U (i.e. $\{u_1, \ldots, u_k\}$), and request a human creator to create system responses for these inputs (i.e. $\{s_1, \ldots, s_k\}$). The strategies for selecting $\{u_1, \ldots, u_k\}$ are proposed in Section 3. Then, Lines 7 and 8 update example base D by adding created example pairs $\{\langle u_1, s_1 \rangle, \ldots, \langle u_k, s_k \rangle\}$, and reconstruct S with the updated dialog examples. Finally, Line 9 evaluates the performance of the updated S on the test data T.

3 Input selection strategies

The performance of Algorithm 1 heavily depends on how we select input from U to present to human creators (i.e. Line 5). In this section, we propose 2 strategies (**DUnc** and **PUnc**) for selecting effective input from U to present to a human creator. These methods are based on the intuition that we can expect that covering examples that are not yet well covered by the database will compensate for the current database's weaknesses. This strategy will improve the dialog system's ability to respond to a variety of inputs. This intuition is well known as uncertainty sampling in general active learning research [25].

Our proposed strategies select *the user input that is different from the dialog examples in D*. In this strategy, the similarity between 1) input u_j in U and 2) example input u_i in D are calculated as the score of u_j:

$$s(u_j) = 1 - \max_{u_i \in D} \frac{\mathrm{w}(u_j) \cdot \mathrm{w}(u_i)}{|\,\mathrm{w}(u_j)\,| \cdot |\,\mathrm{w}(u_i)\,|}. \tag{5}$$

After all of u_j in U are scored according to Eq. (5), k inputs in U are presented to the human creator according to two different sampling methods:

DUnc: samples the k inputs with the highest score.

PUnc: samples k inputs according their probabilities calculated by:

$$\frac{s(u_j)}{\sum_{u_k \in U} s(u_k)}. \tag{6}$$

Queries selected by DUnc are strongly biased by Eq. (5) because of deterministic sampling. To examine the effect of this bias, we additionally introduce the strategy based on probabilistic sampling (i.e., PUnc), which falls halfway between random sampling and the deterministic strategy.

4 Experiments

4.1 Experimental setup

To evaluate the input selection strategies proposed in Section 3, we performed an experimental evaluation using a simulated active learning setup[2]. For simulation, we divided the dialog corpora into initial dialog examples D_0, simulated input logs U and test dialogs T. Each set has oracle pairs of inputs and system responses. Each selection strategy selects the most appropriate input u from input log U to update the dialog manager, and the human creator simulator returns the oracle system response corresponding to the given input. For evaluation, in addition to 2 strategies proposed in Section 3, we use a **Random** baseline, which randomly selects input to present to the human creator simulator. We compared these 3 strategies based on system performance calculated with similarity between the system response and oracle system response (i.e. Eq. (4) in Section 2). Starting with different initial dialog examples D_0, we repeated this evaluation 50 times for each strategy, and used the

[2] Source files to replicate these experiments are available: https://github.com/TakuyaHiraoka/Active-Learning-for-Example-based-Dialog-Systems

average of system performance for each number of annotated examples. Note that, in simulations, if input in U already overlaps an example input in example database D, the overlapping input is deleted from U.

To ensure the portability of the proposed strategies, we prepared 6 simulation domains, which are based on different dialog corpora open to the public:

BusInfo: Human-system dialog for bus information retrieval [26].
Restaurant: Human-system dialog in restaurant information retrieval [27].
Tourist: Human-system dialog in tourist information retrieval [28].
ChatBot: Dialogs between humans and a Japanese chatbot [29].
CleverBot: Dialogs between humans and an English chatbot[3].
Movie: Human-human dialog in movies and television [8, 30].

We calculated several properties of each domain as shown in Table 1. First we calculate complexity of the domain based on the entropy of the input $Hu_d = E[-log_2 P(u_d)]$ and the system response $Hs_d = E[-log_2 P(s_d)]$, where u_d represents all inputs in domain d, s_d represents all system responses in d, and P represents unigram probability. These entropies quantify how much variety exists in the input or system response appearing in each domain. In this research, we use Kylm[4] to calculate each entropy. Furthermore, we describe input types (2nd column of Table 1) in each domain. In the domain annotated with SDF, semantic and discourse features[5] are used as the input to the system. In addition, in the domain annotated with BOW, a bag of words in user utterance is used as the input to the system.

4.2 Experimental results and additional analysis

Figure 3 shows the learning curves of each strategy, where the system performances of each strategy is plotted at each training epoch. In addition, to summarize the performance of each strategy, we define the improvement of the uncertainty-based

Table 1 Input type, number of examples, and entropy of each simulation domain.

Domain	Input type	$\|D_0\|$	$\|T\|$	$\|U\|$	Hu_d	Hs_d
BusInfo	SDF	661	7000	6175	8.661	7.691
Restaurant	SDF	505	6500	6488	6.397	7.371
Tourist	SDF	418	8000	7534	6.518	7.832
ChatBot	BOW	500	6500	6363	6.297	6.863
CleverBot	BOW	248	9500	9593	8.677	8.415
Movie	BOW	753	8500	8540	8.39	8.402

[3] We use dialog logs collected from http://www.cleverbot.com/j2convbydate-page1
[4] http://www.phontron.com/kylm/
[5] In our research, we use previous dialog act and slot filling status [4] as semantic and discourse features

strategies over Random in domain d as average ratio of performance of uncertainty-based strategies to that of Random:

$$AR_{d,sys} = \frac{1}{E_d} \sum_{e=1}^{E_d} \frac{p_{d,sys}}{p_{d,ran}}, \tag{7}$$

where E_d represents the maximum training epoch at domain d, $p_{d,sys}$ represents the system performance of the selected strategy at d according to Eq. (4), and $p_{d,ran}$ represents the score of Random calculated in the same manner as $p_{d,sys}$.

The experimental result in Figure 3 indicate that *proposed strategies based on uncertainty (DUnc, PUnc) can be a good alternative to Random in some domains*. Performances of DUnc were better than those of Random in some domains (BusInfo and ChatBot), and especially its performance in BusInfo was much better than others. In addition, performance of PUnc was equal to or better than Random in all domains except for CleverBot. One of the reasons why these strategies outperformed Random in some domains is that these strategies tend not to select redundant inputs as queries to the dialog creator simulator. For example, in ChatBot, Random selected "ところで、今何をしているのですか？ (By the way, what are you doing now?)" and "何していますか？ (What are you doing now?)". These inputs are not perfectly overlapped, but not very different, and thus we can not expect system performance to increase efficiently by creating system responses for these inputs. Note that the performance of each strategy is dependent on the domain, and these are not necessarily better than that of Random.

Additional analysis indicated that *the DUnc and PUnc strategies can be expected to achieve better performance than Random in domains where the complexity of the system utterance is low.* This was made clear by a correlation analysis between 1) the improvement of uncertainty based strategies according to Eq. (7) and 2) properties of each domain (described in Table 1). The result of this analysis (Figure 4) indicated that there is a strong correlation between the improvement of DUnc and PUnc from Random and entropy of system response Hs_d. If entropy of the system response is high, the system response may be different even if inputs are similar. For example, in CleverBot where the entropy of the system response is high, the system response to input "You're a stupid" is "No you are." whereas the response to the input "You are stupid bot" is "We are the robots". In such a case, considering only information of the input is not enough, and information about the system response is also required to make the proposed strategies be a good alternative to Random.

5 Conclusion

In this paper, we applied active learning to construct example-based dialog managers efficiently. To reduce human effort in creating example bases, we proposed an active learning-based framework, and proposed strategies for selecting input to present to a human creator to create dialog examples. Then, we performed evalua-

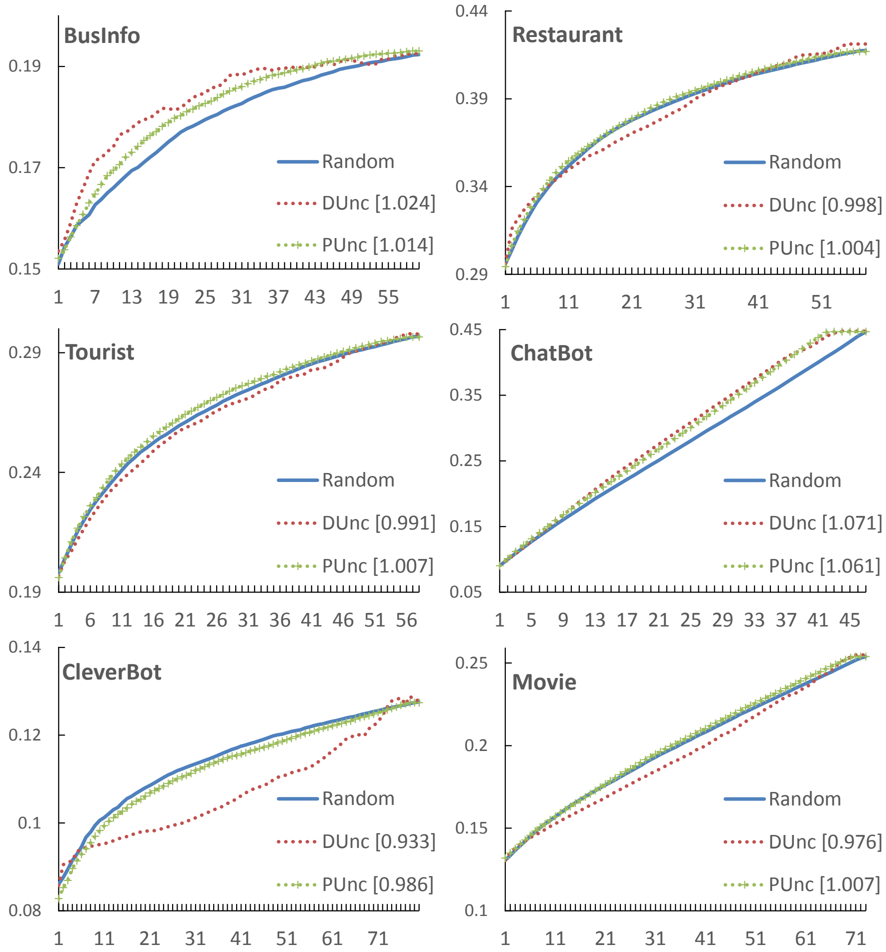

Fig. 3 Learning curves for each input selection strategy in six domains. In each figure, the vertical axes indicate similarity between system responses and oracle responses according to Eq. (4), and the horizontal axes indicate number of training epochs (i.e. e in Algorithm 1). In each training epoch 100 inputs are selected (i.e., $k = 100$). Values at the right side of each label in the legend represent the improvement of uncertainty based strategies from Random according to Eq. (7).

tion based on simulation in 6 different domains. Experimental results and analysis indicated that 1) proposed strategies based on uncertainty can be a good alternative to Random in some domains and 2) these strategies (DUnc, PUnc) can be expected to achieve better performance than Random in domains where the uncertainty of the system utterance is low.

As future work, we plan to propose query selection strategies for domains where system responses are complex, and evaluate with a real human creator. Furthermore, we plan to expand our active learning framework to be more general and cover other response generation frameworks [22, 23, 24].

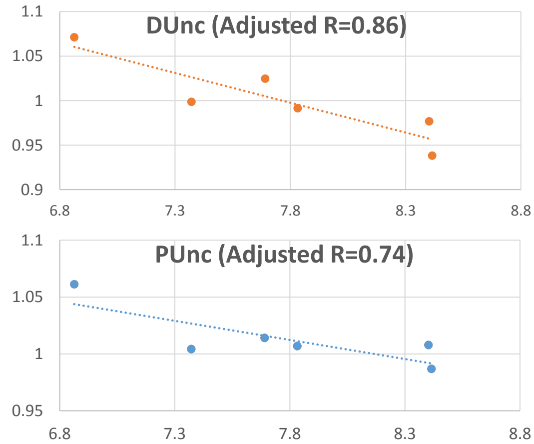

Fig. 4 Correlation between performance $AR_{d,sys}$ and entropy of the system response Hs_d. The top figure shows the case of $sys = DUnc$, and the bottom figure represents the case of $sys = PUnc$. In each figure, the vertical axis represents $AR_{d,unc}$, the horizontal axis represents Hs_d, and each dot represent the tuple $(Hs_d, AR_{d,unc})$ for d.

Acknowledgements Part of this research was supported by JSPS KAKENHI Grant Number 24240032, and the Commissioned Research of National Institute of Information and Communications Technology (NICT), Japan.

References

1. Hiroya Murao, Nobuo Kawaguchi, Shigeki Matsubara, and Yasuyoshi Inagaki, "Example-based query generation for spontaneous speech," *Proceedings of ASRU*, 2001.
2. Hiroya Murao, Nobuo Kawaguchi, Shigeki Matsubara, Yukiko Yamaguchi, and Yasuyoshi Inagaki, "Example-based spoken dialogue system using WOZ system log," *Proceeding of SIGDIAL*, 2003.
3. Ryuhei Nisimura, Yohei Nishihara, Ryosuke Tsurumi, Akinobu Lee, Hiroshi Saruwatari, and Kiyohiro Shikano, "Takemaru-kun: Speech-oriented information system for real world research platform," *Proceedings of LUAR*, 2003.
4. Cheongjae Lee, Sangkeun Jung, Seokhwan Kim, and Gary Geunbae Lee, "Example-based dialog modeling for practical multi-domain dialog system.," *Speech Communication*, 2009.
5. Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft, "Retrieval models for question and answer archives," *Proceeding of SIGIR*, 2008.
6. Rafael E. Banchs and Haizhou Li, "Iris: a chat-oriented dialogue system based on the vector space model," *Proceedings of ACL*, 2012.
7. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura., "Improving the robustness of example-based dialog retrieval using recursive neural network paraphrase identification," *Proceedings of SLT*, 2014.
8. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura, "Utilizing human-to-human conversation examples for a multi domain chat-oriented dialog system," *Transactions of IEICE*, 2014.

9. Burr Settles, “Active learning literature survey,” *Computer Sciences Technical Report 1648, University of Wisconsin-Madison*, 2011.
10. Gokhan Tur, Robert E. Schapire, and Dilek Hakkani-Tur, “Active learning for spoken language understanding,” *Proceedings of ICASSP*, 2003.
11. Gokhan Tur, Dilek Hakkani-Tur, and Robert E. Schapire, “Combining active and semi-supervised learning for spoken language understanding,” *Speech Communication*, 2005.
12. Bjorn Gamback, Fredrik Olsson, and Oscar Tackstrom, “Active learning for dialogue act classification,” *Proceedings of INTERSPEECH*, 2011.
13. Jason D. Williams, Nobal B. Niraula, Pradeep Dasigi, Aparna Lakshmiratan, Carlos Garcia Jurado Suarez, Mouni Reddy, and Geoff Zweig, “Rapidly scaling dialog systems with interactive learning,” *Proceedings of IWSDS*, 2015.
14. Dilek Hakkani-Tur, Giuseppe Riccardi, and Gokhan Tur, “An active approach to spoken language processing,” *ACM Transactions on Speech and Language Processing*, 2015.
15. Fabrizio Ghigi, Vicent Tamarit, Carlos-D. Martinez-Hinarejos, and Jose-Miguel Benedi, “Active learning for dialogue act labelling,” *Lecture Notes in Computer Science*, 2011.
16. Pierre Gotab, Frederic Bechet, and Geraldine Damnati, “Active learning for rule-based and corpus-based spoken language understanding models,” *Proceedings of ASRU*, 2009.
17. Dong Yu, Balakrishnan Varadarajan, Li Deng, and Alex Acero, “Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion,” *Computer Speech and Langueage*, 2010.
18. Giuseppe Riccardi and Dilek Hakkani-Tur, “Active learning: Theory and applications to automatic speech recognition,” *Speech and Audio Processing*, 2005.
19. François Mairesse, Milica Gašić, Filip Jurčíček, Simon Keizer, Blaise Thomson, Kai Yu, and Steve Young, “Phrase-based statistical language generation using graphical models and active learning,” *Proceedings of ACL*, 2010.
20. Milica Gasic and Steve Young, “Gaussian processes for POMDP-based dialogue manager optimisation,” *Audio, Speech, and Language Processing*, 2014.
21. Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman, “Mining of massive datasets,” *Cambridge University Press*, 2014.
22. Alessandro Sordoni, Michel Galle, Michael Auli, Chris Brockett, Yangfeng Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan, “A neural network approach to context-sensitive generation of conversational responses,” *Proceedings of NAACL*, 2015.
23. Alan Ritter, Colin Cherry, and William B. Dolan, “Data-driven response generation in social media,” *Proceedings of EMNLP*, 2011.
24. Andrew Shin, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura, “Context-dependent automatic response generation using statistical machine translation techniques,” *Proceedings of NAACL*, 2015.
25. David D. Lewis and William A. Gale, “A sequential algorithm for training text classifiers,” *Proceedings of SIGIR*, 1994.
26. Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black, “The dialog state tracking challenge.,” *Proceedings of SIGDIAL*, 2013.
27. Matthew Henderson, Blainse Thomson, and Jason Williams, “The second dialog state tracking challenge,” *Proceedings of SIGDIAL*, 2014.
28. Matthew Henderson, Blainse Thomson, and Jason Williams, “The third dialog state tracking challenge,” *Proceedings of SLT*, 2014.
29. Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami, “Towards taxonomy of errors in chat-oriented dialogue systems,” *Proceedings of SIGDIAL*, 2015.
30. Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura., “Conversation dialog corpora from television and movie scripts,” *Proceedings of O-COCOSDA*, 2014.