Dialog System Using Real-Time Crowdsourcing and Twitter Large-Scale Corpus

Fumihiro Bessho, Tatsuya Harada, Yasuo Kuniyoshi
The University of Tokyo
Department of Mechano-Informatics
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{bessho, harada, kuniyosh}@isi.imi.i.u-tokyo.ac.jp

Abstract

We propose a dialog system that creates responses based on a large-scale dialog corpus retrieved from Twitter and real-time crowdsourcing. Instead of using complex dialog management, our system replies with the utterance from the database that is most similar to the user input. We also propose a real-time crowdsourcing framework for handling the case in which there is no adequate response in the database.

1 Introduction

There is a lot of language data on the Internet. Twitter offers many APIs to retrieve or search post status data, and this data is frequently used in research, such as in stock market prediction (Bollen et al., 2011), the spread of information through social media (Bakshy and Hofman, 2011), and representations of textual content(Ramage et al., 2010). Several models for conversation using Twitter data (Ritter et al., 2010; Higashinaka et al., 2011) have been proposed because of the data's vast size and conversational nature.

Kelly (2009) previously showed that 37% of English tweets are Conversational, of which 69% are two-length (one status post and a reply). In our analysis of over 2.5 million tweets, 37.5% of all Japanese tweets are Conversational, which matches Kelly's data. However, less than 58.3% of these are two-length tweets.

Many chat bots are rule-based, which requires a lot of human effort to create or add new rules. For example, A.L.I.C.E (Wallace, 2009), which won the Loebner Prize three times, creates responses based on a dialog strategy database written in a markup language named AIML. Recently, some other chat bots based on a large-scale dialog corpus have been proposed[1,2].

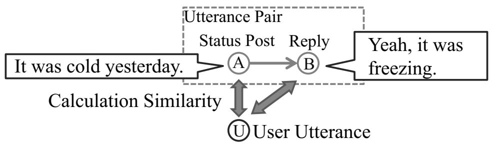


Figure 1: Utterance pair.

In this paper, we propose a novel dialog system (chat bot) that uses real-time crowdsourcing and Twitter large-scale corpus. We evaluate response selection methods based on positive/negative example to judge if each feature could be exploited to judge similarity between uterrances.

2 Method

2.1 Overview

We create an "Utterance Pair" database as shown in Figure 1. Each pair is composed of an utterance (Figure 1, A) and a reply to the utterance (Figure 1, B). Our approach for creating responses is simple and is illustrated in Figure 2. For each user input, the system searches the utterance-pair database for the pair of which the tweet (Figure 1, A) is most similar to that input. The reply contained in this pair (Figure 1, B) forms the system's response to the user's input.

[1]Jabberwacky: http://www.jabberwacky.com
[2]Cleverbot: http://www.cleverbot.com

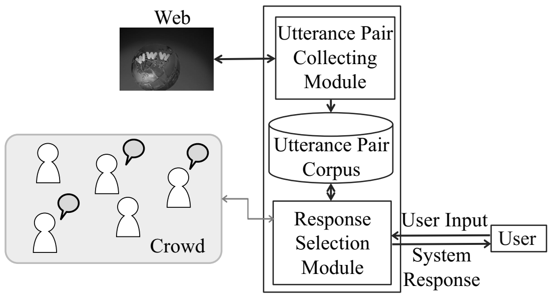


Figure 2: System overview.

If the system cannot find a post that is sufficiently similar to the user's input, then it "outsources" the response to another user.

To build the conversation database, we collected 1.2 million utterance-pairs from the microblogging service, Twitter. We fetched public timeline data using the Streaming API [3], and then looked for tweets which were written in Japanese [4] and had an in-reply-to field. We followed the replies using the REST API [5].

Raw posts from Twitter included *mentions* (the symbol @ followed by a user name), *quotes* (written with letters "RT"), *hashtags* (a word preceded by the symbol #), and *URLs*. We filtered away this information using regular expressions. Unlike English tweets, in Japanese users placed hashtags at the end of their tweet, separated from the bodytext, making the deletion of hashtags feasible.

2.2 Method for the retrieval of Similar Utterance-pairs

In this section, we define a similarity measure between user input and each utterance data in the database. Each utterance in the database is analyzed by a morphological analyzer after Twitter-specific representations are eliminated as mentioned in Section 2.1. Analyzed data are filtered based on part-of-speech (POS) tags. In this paper we only extract noun, verb and interjection, and because many Japanese tweets include emoticons which cannot be tagged correctly by the morphological analyzer we used. We filtered out emoticons using a key-character-filter.

These documents (tweets) were then converted into document vectors. For a document d_i, the vector element corresponding to word w_j is represented as

$$x_{i,j} = \frac{tf_{i,j}}{n_j}, \tag{1}$$

where $tf_{i,j}$ represents the number of times w_j appears in d_i (term frequency), and n_j represents the length of d_i.

The similarity between two documents is calculated by taking the inner product of the two document vectors, that is

$$Similarity(d_a, d_b) = \boldsymbol{x}_a^{\mathrm{T}} \boldsymbol{x}_b. \tag{2}$$

2.3 Real-Time Crowdsourcing

We propose to integrate the dialog system with "real-time crowdsourcing". When the system fails to find an adequate response to a user input, in other words, when the similarity score of the most similar tweet in the database is below a certain threshold, the system relegates the user's input to other users (crowd). The original user input is a tweet to the chat bot and therefore includes the system's name as the target user. In our experiment, the system exchanges its own name to the address of the crowd and utters the tweet to the crowd. If a crowd member responds to the system before a preset timeout period, the system uses the crowd member's reply as a response to the original user. One of the advantages of this method is that people in the crowd do not know that they are part of a crowd; instead, they think they are being addressed by the system. The original user also thinks (s)he is talking with the system. We implemented this real-time crowdosourcing framework using a Twitter clone, StatusNet[6] as an interface (see Figure 5).

3 Evaluation

We prepared 90 user input examples and extracted 20 utterance-pairs (utterance and responses in the database retrieved from Twitter) per user input, so that a total of 1,800 of triples (a user input and an

[3]`https://dev.twitter.com/docs/streaming-api`

[4]We assume tweets as Japanese-written which are written by users who set Language as Japanese.

[5]`https://dev.twitter.com/docs/api`

[6]`http://status.net/`

utterance pair) were included in our sample. Thirty subjects evaluated the naturalness and versatility of the responses. Each subject evaluated 600 triples. We note that subjects saw only the user input and response in an utterance pair (B in Figure 1), and were not shown A in Figure 1 in the survey sheets. In this paper, versatility of a response corresponds to the number of utterances to which it was rated as sensible response (e.g., "What do you mean?" can serve as a response to almost any input, and is therefore highly versatile). Michael (1994) points out that chat bots have many tricks to fool people, and providing a versatile answer is one of them. We believe our system can avoids versatile answers by using a large-scale database.

In the following, we describe how we evaluate each scoring function (which takes a triplet as an input and the score as the output) using positive/negative learning data. We treat scoring functions as classifiers, that is, when the function receives a triplet as input, we assume that the function judges the triplet as positive data if the output score is above a certain threshold and negative data if it is below it.

Triplets collected in the survey were divided into positive and negative triplets. We consider an utterance pair to be *positive* if it is judged as natural by more than 7 out of 10 subjects and versatile by less than 7 out of 10 subjects. All else were considered *negative* triplets.

The ROC curve is used to illustrate the performance of the classifiers. It is a two-dimensional graph in which true positive rate is plotted on the Y axis and false positive rate is plotted on the X axis. Here, the true positive rate (r_{TP}) and false positive rate (r_{FP}) are given by

$$r_{TP} = \frac{\text{Positives correctly classified}}{\text{Total positives}}, \quad (3)$$

$$r_{FP} = \frac{\text{Negatives incorrectly classified}}{\text{Total negatives}}. \quad (4)$$

The area under the curve (AUC), or the area under the ROC curve, was used to measure classifier performance. A random classifier has an AUC of 0.5, and ideal classifier has an AUC of 1.0. We applied a number of scoring functions to the triples, and then calculated the AUC for each function (classifier) for validation. We chose scoring functions which

Calculate similarity with A or A+B.	A+B
Use tf?	YES
Use idf?	NO
Eliminate Twitter-specific representations?	YES
Filter POS?	YES

Table 1: Scoring function we chose.

- calculate similarity only with A in Figure 1, or A and B in Figure 1,
- use term frequency (tf) when the document vector is calculated, or not,
- use inverse document frequency (idf) when the document vector is calculated, or not,
- eliminate Twitter-specific representations (see Section 2.1) or not,
- normalize by character count or not,
- filter POS or not.

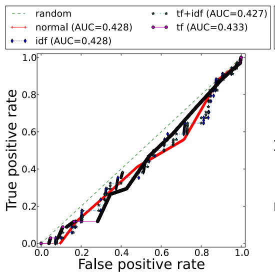
We compared a total of 64 (=2^6) scoring functions. Figure 3 illustrates some or our results. As it shows, when only Twitter-specific expressions are filtered, classifier performance is similar to a random classifier. The addition of word count normalization and POS filter improved to classification performance. This is because longer utterances normally include more specific information, so that the topic is more prone to be missed during the response selection process. Adverbs (e.g. "very") or particles (corresponds preposition in English, e.g. "as") had little effect on the context of an utterance, so POS filtering acts as noise elimination. With respect to tf and idf, the effect of tf varied widely, and idf hindered classification performance (c, d, g, h).

We chose the scoring function with the best performance (see Table 1 for details), of which the AUC is 0.803.

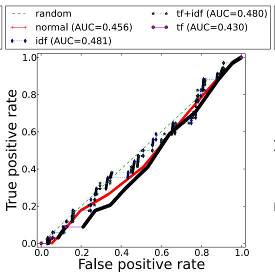
4 Conclusions and Future Work

In this paper, we proposed a new dialog system based on real-time crowdsourcing and a large-scale database which is collected from the web automatically. We also evaluated scoring functions based on positive/negative utterance pairs.

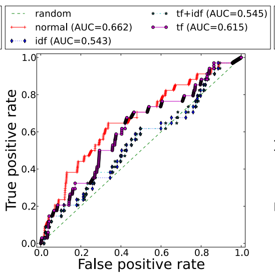
In future work, we will keep on enlarging our utterance pair corpus, and conduct the same experi-



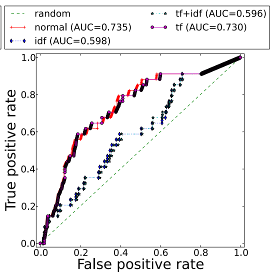
(a) normal (A)



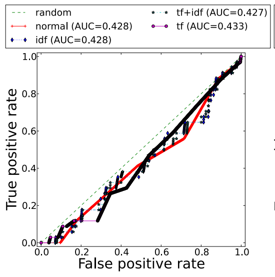
(b) eliminated (A)



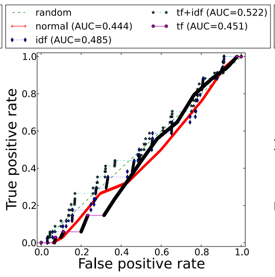
(c) eliminated + normalized (A)



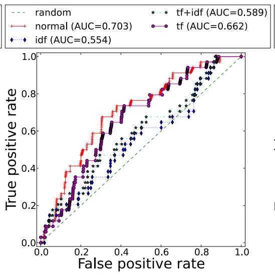
(d) eliminated + normalized + POS filter (A)



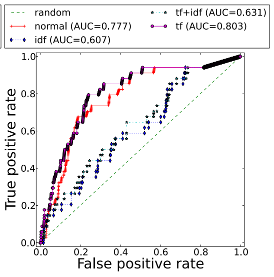
(e) normal (A+B)



(f) eliminated (A+B)



(g) eliminated + normalized (A+B)



(h) eliminated + normalized + POS filter (A+B)

Figure 3: ROC curve for each scoring function. In each graph there are 4 lines, and each line represents whether tf and idf are used to calculate the document vector. Only A in Figure 1 is treated in the first line (a, b, c, d), whereas A and B is considered in the bottom (e, f, g, h). Normal, eliminated, normalized, POS filter mean doing nothing, twitter-specific description is eliminated, normalized by character count, considering only specified POS, respectively.

ments as in this paper on the larger database. We will also use more than two-length tweets for the database. We believe that this approach could lead to a context-aware dialog system (e.g., suppose the user gives "Why?" as the response to the system's utterance "I like dogs.". Our current system cannot handle the topic, "dog", however a system can do so by considering 3 utterances).

References

Eytan Bakshy and Jake M. Hofman. 2011. *Everyone's an Influencer: Quantifying Influence on Twitter*. In Proceedings of International Conference on Web Search and Data Mining.

Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. *Twitter Mood Predicts the Stock Market*. Journal of Computational Science.

Ryuichiro Higashinaka, Noriaki Kawamae, Kugatsu Sadamitsu, and Yasuhiro Minami. 2011. *Building a Conversational Model from Two-Tweets*. IEEE Workshop on Automatic Speech Recognition and Understanding.

Ryan Kelly. 2009. *Pear Analytics Twitter Study (Whitepaper)*.

Michael L. Mauldin. 1994. *Chatterbots, TinyMuds, and the Turing Test Entering the Loebner Prize Competition*. In Proceedings of the National Conference on Artificial Intelligence.

Daniel Ramage, Susan Dumais, and Dan Liebling. 2010. *Characterizing Microblogs with Topic Models*. In Proceedings of the International AAAI Conference on Weblogs and Social Media.

Alan Ritter, Colin Cherry, and Bill Dolan. 2010. *Unsupervised Modeling of Twitter Conversations*. In Proceedings of North American Chapter of the Association for Computational Linguistics - Human Language Technologies.

Richard S. Wallace. 2009. *The Anatomy of A.L.I.C.E.*. Parsing the Turing Test.

USER	こんにちは (konnitiha: Hello)
SYSTEM	こんにちは !! こんにちは !! こんにちは !! (konnitiha!! konnnitiha!! konnnitiha!!: Hello!! Hello!! Hello!!)
USER	今日も寒いですよね。 (kyou-mo-samui-desu-yone: It's cold today, isn't it?)
SYSTEM	だ、大丈夫ゼロ？上着を羽織って、暖房をつけてみたりした？ (da, daijobu-zero? uwagi-wo-haotte, danbou-wo-tuketemitari-sita?: Are you ok? How about putting on a jacket, and turning on the heater?))
USER	暖房つけてるよ。(danbou-tuketeru-yo.: I've already turned on the heater.)
SYSTEM	暖房ないけどあったかい (＊´〜`＊) (danbou-nai-kedo-attakai <emoticon>: There is no heater(in this room), but it's warm <emoticon>)

USER	なんだか熱っぽい。(nandaka-netuppoi: I feel feverish.)
SYSTEM	大丈夫？ (daijoubu?: Are you ok?)
USER	インフルエンザに感染してないといいけど。(Infuruenza-ni-kannsenn-shitenai-to-iikedo.: I hope I'm not ill with flu.)
SYSTEM	インフルエンザ用の効ウィルス剤は感染から 48 時間以内にしかあまり効果がない罠 (infuruenza-you-no-kouuiruzuzai-ha-kannsenn-kara-48jikanninai-ni-shika-amari-kouka-ga-nai-wana: Antiviral drug for flu only works to those who got infected in 48 hours.)
USER	マジか。(majika.: really?)
SYSTEM	寝ろよ！ (neroyo!: You should sleep.)
USER	おとなしく寝ます。(otonashiku-nemasu.: I'll take it easy and sleep.)
SYSTEM	おとなしくするべきだ (otonashiku-suru-beki-da: You should do so.)

Table 2: Dialog examples

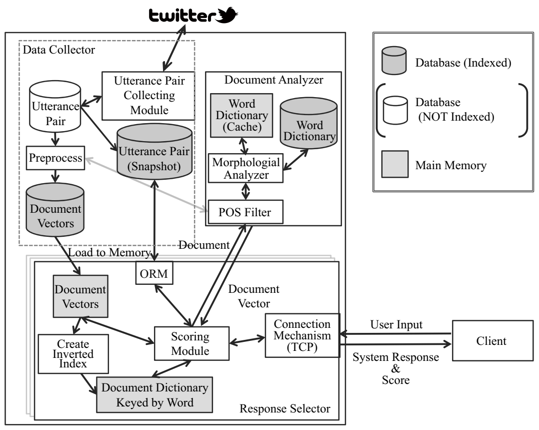
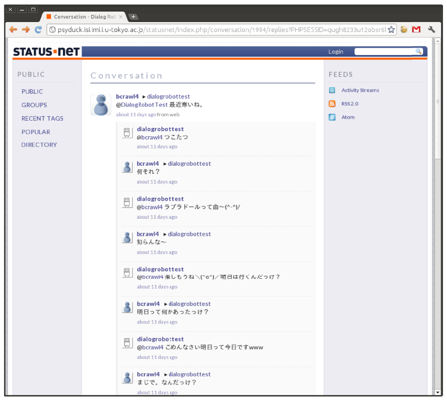


Figure 4: System Implementation.


Conversation - Dialog Rob

psyduck.isi.imi.i.u-tokyo.ac.jp/statusnet/index.php/conversation/1994/replies?PHPSESSID=qugh8233u12obsr6l

STATUS•net Login

PUBLIC
PUBLIC
GROUPS
RECENT TAGS
POPULAR
DIRECTORY

Conversation

bcrawl4 ▸ dialogrobottest
@DialogRobotTest 最近寒いね。
about 11 days ago from web

dialogrobottest
@bcrawl4 つこたつ
about 11 days ago

bcrawl4 ▸ dialogrobottest
何それ？
about 11 days ago

dialogrobottest
@bcrawl4 ラブラドールって曲〜(^-^)/
about 11 days ago

bcrawl4 ▸ dialogrobottest
知らんな〜
about 11 days ago

dialogrobottest
@bcrawl4 楽しもうね`＼(^o^)／'明日は行くんだっけ？
about 11 days ago

bcrawl4 ▸ dialogrobottest
明日って何かあったっけ？
about 11 days ago

dialogrobottest
@bcrawl4 ごめんなさい明日って今日ですwww
about 11 days ago

bcrawl4 ▸ dialogrobottest
まじで。なんだっけ？
about 11 days ago

FEEDS
Activity Streams
RSS 2.0
Atom

Figure 5: System implementation on StatusNet.

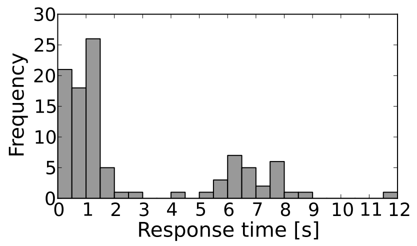


Figure 6: System response time distribution. (datasize = 1,154,621)