

Syntactic Filtering and Content-Based Retrieval of Twitter Sentences for the Generation of System Utterances in Dialogue Systems

Ryuichiro Higashinaka, Nozomi Kobayashi, Toru Hirano,
Chiaki Miyazaki, Toyomi Meguro, Toshiro Makino and Yoshihiro Matsuo

Abstract Sentences extracted from Twitter have been seen as a valuable resource for response generation in dialogue systems. However, selecting appropriate ones is difficult due to their noise. This paper proposes tackling such noise by syntactic filtering and content-based retrieval. Syntactic filtering ascertains the valid sentence structure as system utterances, and content-based retrieval ascertains that the content has the relevant information related to user utterances. Experimental results show that our proposed method can appropriately select high-quality Twitter sentences, significantly outperforming the baseline.

1 Introduction

In addition to performing tasks [19], dialogue systems should be able to perform open-domain conversation or chat in order for them to look affective and to build social relationships with users [2]. Chat capability also leverages the usability of

R. Higashinaka (✉) · N. Kobayashi · T. Hirano · C. Miyazaki · T. Makino · Y. Matsuo
NTT Media Intelligence Laboratories, Kanagawa, Japan
e-mail: higashinaka.ryuichiro@lab.ntt.co.jp

N. Kobayashi
e-mail: kobayashi.nozomi@lab.ntt.co.jp

T. Hirano
e-mail: hirano.tohru@lab.ntt.co.jp

C. Miyazaki
e-mail: miyazaki.chiaki@lab.ntt.co.jp

T. Makino
e-mail: makino.toshiro@lab.ntt.co.jp

Y. Matsuo
e-mail: matsuo.yoshihiro@lab.ntt.co.jp

T. Meguro
NTT Communication Science Laboratories, Kyoto, Japan
e-mail: meguro.toyomi@lab.ntt.co.jp

task-oriented dialogue systems because real users do not necessarily utter only task-related (in-domain) utterances but also chatty utterances [17]; such utterances, if not handled correctly, can cause misunderstandings.

One challenge facing an open-domain conversational system is the wide variety of topics in user utterances. Conventional methods have used hand-crafted rules, but the coverage of topics is usually very limited [20]. To increase the coverage, recent studies have exploited the web, typically Twitter, to extract and use sentences for response generation [1, 15]. However, due to the nature of the web, such sentences are likely to be negatively affected by noise.

Heuristic rules have been proposed by Inaba et al. [10] to filter inappropriate Twitter sentences, but since their filtering is performed on the word level, their filtering capability is very limited. To overcome this limitation, this paper proposes syntactic filtering and content-based retrieval of Twitter sentences; syntactic filtering ascertains the validity of sentence structures and content-based retrieval ascertains that the extracted sentences contain information relevant to user utterances.

In what follows, Sect. 2 covers related work. Section 3 explains our proposed method in detail. Section 4 describes the experiment we performed to verify our method. Section 5 summarizes the paper and mentions future work.

2 Related Work

Conventional approaches to open-domain conversation have heavily depended on hand-crafted rules. The early systems such as ELIZA [21] and PARRY [3] used heuristic rules derived from psycho-linguistic theories. Recent systems at the Loebner prize (a chat system competition) typically use tens of thousands of hand-crafted rules [20]. Although such rules enable high-quality responses to expected user utterances, they fail to respond appropriately to unexpected ones. In such cases, systems tend to utter innocuous (fall-back) utterances or change topic, which often lowers user satisfaction.

To overcome this problem, recent studies have used the web for response generation. For example, Shibata et al. and Yoshino et al. used sentences in web-search results for response generation [15, 22]. To make utterances more colloquial and suitable for conversation, instead of web-search results, Twitter has become the recent target for sentence extraction [1]. Although extracting sentences from the web can deal with a wide variety of topics in user utterances, due to the web's diversity, the extracted sentences are likely to contain noise.

To suppress this noise, Inaba et al. proposed word-based filtering of Twitter sentences [10]. Their rules filter sentences that contain context-dependent words such as referring/temporal expressions. They also score sentences by using the weights of words calculated from a reference corpus and remove those with low scores. Our motivation is similar to Inaba et al.'s in that we want to extract sentences from Twitter that are appropriate as system utterances, but our work is different in that, in addition

to word-level filters, we also take into account the syntax and the content of Twitter sentences for more accurate sentence extraction.

Although not within the scope of this paper, there are emerging approaches to building knowledge bases for chat systems by using web resources. Higuchi et al. mined the web for associative words (mainly adjectives) to fill in their generation templates [8], and Sugiyama et al. created a database of dependency structures from Twitter to find words for their templates [16]. Statistical machine translation techniques have also been utilized to obtain transformation rules (as a phrase table) from input to output utterances [14]. Although we find it important to create good knowledge bases from the web for generation, since it is still in a preliminary phase and the reported quality of generated utterances is rather low, we currently focus on the selection of sentences.

3 Proposed Method

In this paper, we assume that the input to our method is what we refer to as a *topic word*. A topic word (represented by noun phrases in this paper) represents the current topic (focus) in dialogue and can be obtained from a user utterance or from the dialogue context. We do not focus on the extraction of topic words in this paper; note that finding appropriate topic words themselves is a difficult problem, requiring the understanding of the context.

Under this assumption, our task is to retrieve appropriate sentences from Twitter given a topic word. Our method comprises four steps: preprocess, word-based filtering, syntactic filtering, and content-based retrieval. Note that, in this paper, we assume the language used is Japanese.

3.1 Preprocess

As a preprocess, input tweets are first stripped of Twitter-dependent expressions (e.g., retweeted content and user names with mention markers). Then, the tweets are split into sentences by sentence-ending punctuation marks. After that, sentences that are too short (less than five characters) or too long (more than 30 characters) are removed because they may not be appropriate as colloquial utterances. We also remove sentences that contain no Japanese characters.

3.2 Word-Based Filtering

The sentences that pass the preprocess are processed by a morphological analyzer. The sentences together with their analysis results are sent to the word-based filters. There are three filters:

- (1) **Sentence Fragment Filter** If the sentence starts with sentence-end particles, punctuation marks, or case markers (Japanese case markers do not appear at the beginning of a sentence), it is removed. If the sentence ends with a conjunctive form of verbs/adjectives (meaning that the sentence is not complete), it is removed. This filter is intended to remove sentence fragments caused mainly by sentence splitting errors.
- (2) **Reference Filter** If the sentence contains pronouns, deixes, or referring expressions such as ‘it’ and ‘that’, it is removed. If the sentence has words related to comparisons (such as more/than) or an anteroposterior relation (such as following/next), it is also removed. If the sentence has words representing reason or cause, it is removed. If the sentence contains relation-related words, such as family members (mother, brother, etc.), it is also removed. Such sentences need to be removed because entities and events being referred to may not be present in the sentence or differ depending on the speaker.
- (3) **Time Filter** If the sentence contains time-related words, such as dates and relative dates, it is removed. If the sentence has verbal suffixes representing past tenses (such as ‘mashita’ and ‘deshita’), it is also removed. Such sentences are associated with certain time points and therefore may not be used independently of the context.

The filters here are similar to those used by Inaba et al. [10] with some extensions, such as the use of tense and relation-related words. The filters are applied to input sentences in a cascading manner. If a sentence passes all the filters, it is sent to syntactic filtering.

3.3 Syntactic Filtering

The sentences are checked with regard to their syntactic structures. This process is intended to ascertain if the sentence is structurally valid as an independent utterance; that is, the sentence is grammatical and has necessary arguments for predicates. For example, “watashi wa iku (I go)” does not have a destination for the predicate “go”, making it an non-understandable utterance on its own.

However, such checks are actually difficult to perform. This is because Twitter sentences are mostly in colloquial Japanese with many omissions of particles and case markers, making it hard to use the rigid grammar of written Japanese for validation. In addition, missing arguments do not necessarily mean an invalid structure because Japanese contains many zero-predicate and zero-pronoun structures. For example, “eiga ni ikitai (want to go to the movies)” does not have a subject for a predicate, but since the sentence is in the desiderative mood, we can assume that the subject is “watashi (I)” and the sentence is thus understandable. The checks need to take into account the types of predicates as well as mood, aspect, and voice, making it difficult to enumerate by hand all the conditions when a sentence can be valid. Therefore, to automatically find conditions when a sentence is valid, we turn to a machine

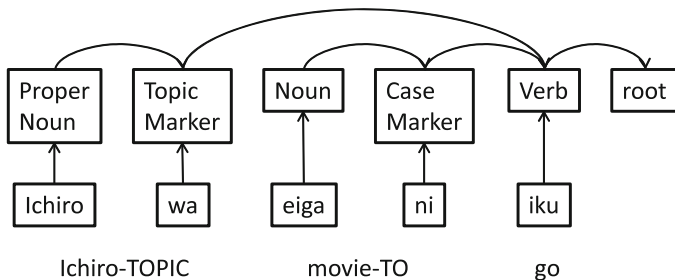


Fig. 1 A word dependency tree for “Ichiro wa eiga ni iku (Ichiro goes to the movies)”. The nodes of base forms and end forms are omitted from illustration because they are exactly the same as word surfaces in this example

learning based approach and use a binary classifier that has been trained from data to determine whether a sentence is valid or invalid on the basis of its structure. Note that the aim of this filtering is NOT to guarantee the “syntactic well-formedness” of sentences since responses need not be syntactically well-formed in “chit-chat” type interactions; here we simply want to remove sentences that are considered invalid from their structures. Below shows how we created the classifier.

3.3.1 Machine Learning Based Classifier

To create the classifier, we first collected Twitter sentences and labeled them as valid (i.e., positive examples) and invalid (i.e., negative examples). Then, we converted the sentences into word dependency trees by using a dependency analyzer in a manner similar to Higashinaka and Isozaki [7]. The trees have part-of-speech tags as main nodes with word surfaces, base forms, and end forms as their daughters (see Fig. 1 for an example). Finally, the trees of negative and positive examples were input to BACT [11], a boosting based algorithm for classifying trees, to train a binary classifier. BACT enumerates subtrees in the input data and uses the existence of the subtrees as features for boosting-based classification. Since subtrees are used as features, syntactic structures are taken into account for classification.

For creating the training data, we sampled 164 words as topic words from our dialogue corpus [13]. Then, for each topic word, we retrieved up to 100 Twitter sentences by using a text search engine that has an index similar to (d) in Table 1 with a content-based retrieval method we describe later (see Sect. 3.4). For the retrieved sentences, an annotator, who is not the author, labeled validity scores on a five-point Likert scale where 1 indicates completely invalid and 5 completely valid. We treated sentences scored 1 and 2 as negative examples and those scored 4 and 5 as positive examples. We did not use sentences scored 3. In total, we created 3880 positive and 1304 negative examples. By using these data, a classifier was learned by BACT.

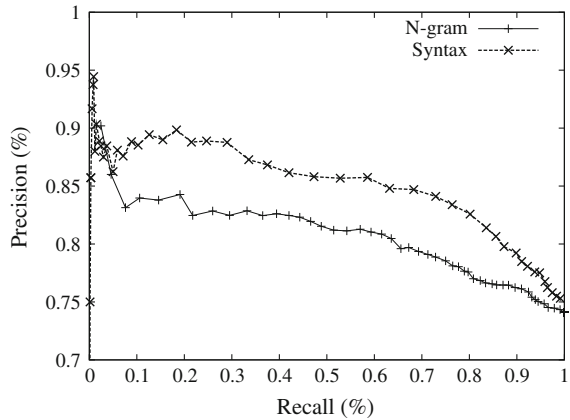
The evaluation was done by using a twofold cross validation, with each fold having examples regarding 82 topic words. Figure 2 shows the recall-precision curves for the

Table 1 Statistics of our Twitter data

| | Number | Retained ratio |
|--|-------------|----------------|
| (a) Number of tweets | 397,288,109 | N/A |
| (b) Number of sentences | 870,471,300 | 100.00 % |
| (c) Number of sentences retained by word-based filtering | 103,655,452 | 11.9 % |
| (d) Number of unique sentences | 53,379,647 | 6.1 % |
| (e) Number of unique sentences retained by the syntactic filtering | 7,907,888 | 0.9 % |

Retained ratio is the ratio of retained sentences over (b)

Fig. 2 Recall-precision curves for N-gram based and syntactic filtering. The graph shows the result for one of the folds in twofold cross validation. The other fold has the same tendency



trained syntactic classifier (Syntax) with a comparison to an N-gram based baseline (N-gram). Here, the baseline uses word and part-of-speech N-grams (unigrams to 5-grams) as features with logistic regression as a training algorithm [4]. The curves show that our trained syntactic filter classifies sentences with good precision. It is also visible that the syntactic filter consistently outperforms the baseline. As a requirement for a filter, low false acceptance is desirable. By a statistical test (a sign-test that compares the number of times the syntactic filter outperforms the N-gram based filter and vise-versa), we confirmed that the syntactic filter has significantly lower false acceptance than the baseline ($p < 0.001$), verifying the usefulness of syntactic information.

3.3.2 Filtering by the Trained Classifier to Create an Index

On the basis of the evaluation result, we decided to use the syntactic classifier (trained with all the examples) to filter input sentences. The sentences that pass this filter are indexed by a text search engine (we use Lucene, see Sect. 4.1) that allows for efficient searching.

3.4 Content-Based Retrieval

Content-based retrieval can retrieve sentences that contain information related to an input topic word. For this, we use a dictionary of *related words*. Related words are the words strongly associated with a topic word. We collect such words from the web and use them to expand the search query so that the retrieved sentences contain such words.

The idea here is inspired by the work of Louis and Newman [12] that uses related words for tweet retrieval, but our work is different in that we allow arbitrary words as input (not just an named-entity type) and use a high-quality dictionary of related words by strict lexico-syntactic patterns, not just a simple word collocation.

3.4.1 Creating a Dictionary of Related Words

We use lexico-syntactic patterns to extract related words. Lexico-syntactic patterns have been successfully used to obtain related words such as hyponyms [6] and attributes [18].

For a given word *W*, we collect noun phrases (NP), adjectives (ADJ), and verbs (V) as related words. For noun phrases, we use a lexico-syntactic pattern similar to that used by Tokunaga [18] and collect attributes of *W*. More specifically, we use the pattern “*W* no NP (*ga* | *wo* | *ni* | *de* | *kara* | *ori* | *e* | *made*)”, corresponding to “NP of *W* Verb” in English. We collect attributes because they form part of a topic word and therefore are likely to be closely related. For adjectives, we use the pattern “*W* (*ga* | *wa*) ADJ”, corresponding to “*W* is ADJ” in English. This pattern retrieves adjectival properties of *W*. For verbs, we use “*W* (*ga* | *wo* | *ni* | *de*) V” where *W* appears in the important argument positions (nominative, accusative, dative, and locative positions) of *V*.

By using the weblogs of 180M articles we crawled, we used the above patterns to extract the related words for all noun phrases in the data. Then, we distilled the results by filtering words that do not correlate well with the entry word (i.e., *W*). We used the log likelihood ratio test (G-test) to determine whether a related word appears significantly more than chance. We retained only the related words that have the G value of over 10.83 (i.e., $p < 0.001$). Finally, the retained words comprise our related word dictionary. The dictionary contains about 2.2M entries. To give a brief example, an entry of “Ramen” (a type of noodle dish) includes noodles, soup, restaurant as NP, delicious, tasty, longing as ADJ, and eat, order, sip, for V.

3.4.2 Retrieval Method

Given a topic word *T*, we search for top-*N* sentences from the index. Here, we score a sentence *S* by the following formula:

$$score(T, S) = \sum_{w \in (rel(T) \cap words(S))} weight(w) \quad (1)$$

Here, ‘rel’ returns the set of related words for T, and ‘words’ returns the set of words contained in S. ‘weight’ returns the G value (we used the logarithm of G value in order to normalize its range) for a word in the related word dictionary. By this formula, the sentences that have many related words are ranked highly, resulting in the retrieval of sentences that are likely to contain just the information related to the topic word. Note that since we assign no weight to non-related words, the formula *relatively* lowers the rank of sentences that contain irrelevant content.

4 Experiment

We performed an experiment to verify our approach. We first created an index of sentences from the Twitter data we crawled. Then, we evaluated the quality of utterances of our proposed method by using human subjects.

4.1 Data

First, we crawled about 400M tweets. Then, we followed the steps of our proposed method to create an index for sentence retrieval. Table 1 shows the statistics of our data. The 870M sentences at the beginning were reduced to 8M sentences after all the filters (including an additional unifying process) had been applied. Here, in an attempt to make our syntactic filter more sensitive to false acceptance, we used 0.005 as a cut-off threshold (default 0.00). We created two indices from the data: one created with (d) and the other with (e). The aim of this is to compare the effectiveness of the syntactic filter in the experiment we describe later. We call the former *the whole index* and the latter *the filtered index*. We used Lucene, which is an open source text search engine, to create the indices.

4.2 Experimental Procedure

We made four systems for comparison: one is the baseline that only uses word-based filtering, and the others are variations of the proposed method. The systems are as follows:

Baseline The whole index is used for sentence retrieval. In ranking the sentences, a vector space model using TF-IDF weighted word vectors is used. This is the default search mechanism in Lucene. This is the condition where there is no syntactic filter or content-based retrieval.

| |
|--|
| amazon, Minatomirai, Iraq, Cocos, Smart-phone, Disney Sea, news, Hashed Beef, Hello Work, FamilyMart, Fuji Television, horror, Pocari Sweat, Mister Donut, mosquito, weather, Kinkakuji temple, accident, Hatsushima, Shinsengumi, fortune-telling, region, local area, Tokyo Bay, pan, Yatsugatake, damage, Kitasenju, Meguro, baseball club, courage |
|--|

Fig. 3 Topic words used for the experiment. The words were originally in Japanese and translated by the authors

Syntax The filtered index is used for sentence retrieval but the content-based retrieval is not used.

Content The whole index is used for sentence retrieval and the content-based retrieval is used.

Content+Syntax The filtered index is used with content-based retrieval. This is the full implementation of our proposed method.

For the morphological analyzer and the dependency analyzer, we used NTT's JTAG [5] and JDEP [9], respectively.

For the evaluation, we first sampled 31 words as topic words (see Fig. 3) from our dialogue corpus [13]. They do not overlap with ones we used for training our syntactic filter. Then, we made each system output three utterances for each topic word. Here, the three utterances are those randomly taken from the top-10 retrieved sentences. We did this process because we considered that it may not be sufficient to evaluate only the top-1 sentence; dialogue systems usually continue on the same topic for a certain number of turns, making it necessary for the systems to be able to create multiple sentences for a given topic. In addition, it is common practice in chat systems that sentences be randomly chosen from a pool of sentences for making variation in utterances. We believe evaluating randomly selected utterances from top-ranked retrieved sentences is appropriate in terms of actual system deployment. By this procedure, we created 93 utterances for each system, for a total of 372 utterances.

We had two judges, who are not any of the authors, subjectively evaluate the quality of the generated utterances (shown with topic words and presented in a randomized order) in terms of (i) understandability (if the utterance is understandable as a response to a topic word) and (ii) continuity (if the utterance makes you willing to continue the conversation on the topic) on a five-point Likert scale, where 1 is the worst and 5 the best. We use averaged understandability and continuity scores to evaluate the systems. In addition to these metrics, we also use a metric that we call (iii) non-understanding rate, which is the rate of lowly rated utterances (scores 1 and 2) in the understandability score over the number of total utterances. Since even a single non-understandable utterance can lead to a sudden breakdown in conversation, we consider this figure to be an important indicator of robustness to keep the conversation on track. Each utterance was judged independently.

Table 2 Averaged understandability scores, continuity scores, and non-understanding rates

| | Baseline | Syntax | Content | Syntax+content |
|------------------------|----------|--------|---------|----------------|
| Understandability | 2.68 | 3.55 | 3.53 | 3.92 |
| Continuity | 2.68 | 3.60 | 3.61 | 4.06 |
| Non-understanding rate | 0.47 | 0.15 | 0.18 | 0.06 |

The averaging was done over all samples given by the two raters for each system. For understanding and continuity scores, the four methods significantly differ in performance ($p < 0.01$) except between Syntax and Content

4.3 Results

Table 2 shows the averaged understandability scores, continuity scores, and non-understanding rates. It can be seen that when the syntactic filtering and the content-based retrieval are used, the performance is the best. Regarding the understandability and continuity scores, statistical tests (Wilcoxon rank sum test with Bonferroni adjustment for multiple comparison) show that the proposed system and the other three systems significantly differ. In fact, scores for the four systems significantly differ except between Syntax and Content, meaning that syntactic filtering and content-based retrieval have their own merits and are complimentary. We can also see that the word-based filtering alone cannot guarantee the quality of selected sentences at all. When we look at the non-understanding rates, we find that Syntax+Content achieves a very low figure of 6 %, suggesting that in most cases the utterances do not lead to a sudden breakdown of dialogue.

Within the utterances that Syntax+Content created, only one utterance scored 1 for understandability:

- (1) aiteru-yoo kitasenju-ni ii yakinikuya-*kara ikou-zee
open-SEP Kitasenju-at good BBQ-restaurant-from go-SEP
‘It’s open. Why don’t we go *from the good BBQ restaurant at Kitasenju’

Here, SEP denotes a sentence-end particle and an asterisk means ungrammatical. This sentence contains two sentences without any punctuation mark in between, and the first sentence has a missing argument and the second sentence has an incorrect predicate-argument structure. The trained syntactic classifier probably failed to detect it as invalid because such a complex combination of errors was not seen in the training data. An increase in the training data could solve the problem.

5 Summary and Future Work

This paper proposed syntactic filtering and content-based retrieval of Twitter sentences so that the retrieved sentences can be safely used for response generation in dialogue systems. Experimental results showed that our proposed method can

appropriately select high-quality Twitter sentences, significantly outperforming the word-based baseline. Our contribution lies in discovering the usefulness of syntactic information in filtering Twitter sentences and in validating the effectiveness of related words in retrieving sentences. For future work, we plan to investigate how to extract topic words from the context and also to create a workable conversational system with speech recognition and speech synthesis.

Acknowledgments We thank Prof. Kohji Dohsaka of Akita Prefectural University for his helpful advice on statistical tests. We also thank Tomoko Izumi for her suggestions on how to write linguistic examples.

References

1. Bessho F, Harada T, Kuniyoshi Y (2012) Dialog system using real-time crowdsourcing and Twitter large-scale corpus. In: Proceedings of the SIGDIAL, pp 227–231
2. Bickmore TW, Picard RW (2005) Establishing and maintaining long-term human-computer relationships. *ACM Trans Comput-Hum Interact* 12(2):293–327
3. Colby KM, Watt JB, Gilbert JP (1966) A computer method of psychotherapy: preliminary communication. *J Nerv Mental Dis* 142(2):148–152
4. Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ (2008) LIBLINEAR: a library for large linear classification. *J Mach Learn Res* 9:1871–1874
5. Fuchi T, Takagi S (1998) Japanese morphological analyzer using word co-occurrence—JTAG. *Proc COLING-ACL* 1:409–413
6. Hearst MA (1992) Automatic acquisition of hyponyms from large text corpora. *Proc COLING* 2:539–545
7. Higashinaka R, Isozaki H (2008) Automatically acquiring causal expression patterns from relation-annotated corpora to improve question answering for why-questions. *ACM Trans Asian Lang Inf Process* 7(2)
8. Higuchi S, Rzepka R, Araki K (2008) A casual conversation system using modality and word associations retrieved from the web. In: Proceedings of the EMNLP, pp 382–390
9. Imamura K, Kikui G, Yasuda N (2007) Japanese dependency parsing using sequential labeling for semi-spoken language. In: Proceedings of the ACL, pp 225–228
10. Inaba M, Kamizono S, Takahashi K (2013) Utterance generation for non-task-oriented dialogue systems using Twitter. In: Proceedings of the 27th annual conference of the Japanese society for artificial intelligence. 1K4-OS-17b-4 (in Japanese)
11. Kudo T, Matsumoto Y (2004) A boosting algorithm for classification of semi-structured text. In: Proceedings of the EMNLP, pp 301–308
12. Louis A, Newman T (2012) Summarization of business-related tweets: A concept-based approach. In: Proceedings of the COLING 2012 (Posters), pp 765–774
13. Meguro T, Higashinaka R, Minami Y, Dohsaka K (2010) Controlling listening-oriented dialogue using partially observable Markov decision processes. In: Proceedings of the COLING, pp 761–769
14. Ritter A, Cherry C, Dolan WB (2011) Data-driven response generation in social media. In: Proceedings of the EMNLP, pp 583–593
15. Shibata M, Nishiguchi T, Tomiura Y (2009) Dialog system for open-ended conversation using web documents. *Informatica (Slovenia)* 33(3):277–284
16. Sugiyama H, Meguro T, Higashinaka R, Minami Y (2013) Open-domain utterance generation for conversational dialogue systems using web-scale dependency structures. In: Proceedings of the SIGDIAL, pp 334–338

17. Takeuchi S, Cincarek T, Kawanami H, Saruwatari H, Shikano K (2007) Construction and optimization of a question and answer database for a real-environment speech-oriented guidance system. In: Proceedings of the Oriental COCOSDA
18. Tokunaga K, Kazama J, Torisawa K (2005) Automatic discovery of attribute words from web documents. In: Proceedings of the IJCNLP, pp 106–118
19. Walker MA, Passonneau R, Boland JE (2001) Quantitative and qualitative evaluation of DARPA communicator spoken dialogue systems. In: Proceedings of the ACL, pp 515–522
20. Wallace RS (2004) The anatomy of A.L.I.C.E. A.L.I.C.E. artificial intelligence foundation, Inc
21. Weizenbaum J (1966) ELIZA-a computer program for the study of natural language communication between man and machine. Commun ACM 9(1):36–45
22. Yoshino K, Mori S, Kawahara T (2011) Spoken dialogue system based on information extraction using similarity of predicate argument structures. In: Proceedings of the SIGDIAL, pp 59–66

Situated Dialog in Speech-Based Human-Computer
Interaction

Rudnicky, A.; Raux, A.; Lane, I.; Misu, T. (Eds.)

2016, VII, 225 p. 71 illus., 41 illus. in color., Hardcover

ISBN: 978-3-319-21833-5