Automatic Utterance Generation in Consideration of Nominatives and Emoticon Annotation

Yusuke Nishio
Graduate School of
Integrated Basic Sciences
Nihon University, Tokyo,
JAPAN

Mirai Miura
Graduate School of
Information Science
Nara Institute of
Science and Technology
Nara, JAPAN

Dongli Han
Department of Information
Science
College of Humanities and
Sciences
Nihon University, Tokyo,
JAPAN
han@chs.nihon-u.ac.jp

Abstract

The demand is increasing recently for non-task-oriented conversation system in various scenes. Previous studies provide various strategies to enrich the methods for generating utterances, thus making the conversation systems or agents appear more interesting. However, most previous works tend to rely on templates and therefore are not able to perform flexible conversation-utterance generation. We propose here in this paper a thorough modification to a previous work to address this problem. Specifically, we introduce an automatic utterance generation in consideration of the embedded structure of sentences based on the principle of nominative maintenance. Moreover, emotion presumption has been implemented to add entertaining elements into the conversation with a user. The experimental results show that our approach proposed in this study has helped improve the performance of a conversation system.

1 Introduction

Non-task-oriented conversation systems have been developed during the past decade. They pay more attention to continuing the conversation by any means rather than the rigorousness of the utterance's content in comparison with task-oriented ones. However, the insufficiency in methods for generating utterances still remains as a critical issue unsolved.

For example, Higuchi et al. concentrate on modalities appearing in human's utterances, and try to incorporate them into the process of utterance generation (Higuchi et al., 2008). Song et al. (Song et al., 2009) and Han et al. (Han et al., 2010) present a strategy to provide new topics for users in a free conversation system at the point the system "considers" that the user has lost interest in the current topic. As just described, most previous studies provide various strategies to enrich the methods for generating utterances, so that the user might feel interested in the system and intend to continue the conversation. However, none of them could escape from the fact that they all generate utterances depending mainly on some particular kinds of templates or augmented templates.

As a case study to cope with this problem, Han et al. develop a free conversation system employing Markov sequences as shown in Figure 1 (Han et al., 2011; Nishio and Han, 2012). They use the topic-word pair extracted beforehand to search the Twitter for snippets that contain a noun in the beginning and a verb or adjective in the end, and then generate an utterance employing a Two-starting-word style Markov connection.

Although this approach has been proven quite effective in promoting the human-like qualities of utterances, a significant problem has been observed simultaneously: Utterance Focus usually spreads around too widely. This seems to come from the nature of the two-starting-word method which tends to generate comparatively long computer-utterances.

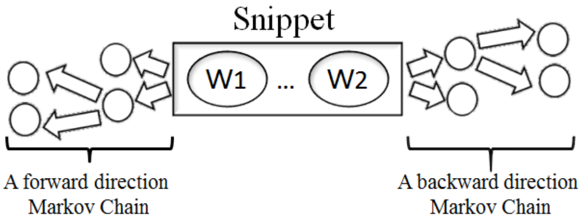
Snippet
W1
...
W2
A forward direction Markov Chain
A backward direction Markov Chain

Figure 1: Processing schema of multiple-word Markov connection

Here we propose a strategy to cope with this issue. Specifically, we incorporate some restrictive rules to improve this situation by limiting the total number of words or characters contained in the generated utterance. Moreover, to add more entertaining elements into the system, we implement an Emoticon Annotating function to assign an emoticon to the utterances.

In this paper, we first describe the theoretical basis and specific steps of the utterance generation process in Section 2 and 3. Then we propose a method to presume possible emotions for an input text in Section 4. Section 5 elaborates the function we have implemented to annotate an emoticon to a computer-generated utterance based on the results of emotion presumption. Finally, we give some experimental results for verifying the effectiveness of our approach in Section 6.

2 Theoretical Basis

As we have mentioned in Section 1, utterance generation in the previous work is conducted based on Markov chains. The snippet is extracted randomly instead of using any restriction rule. As a result, long utterances are easily generated whose focuses tend to spread around too widely. To address this issue, we have to figure out some strategies to limit the total number of words or characters contained in the generated utterance. In this section, we introduce two concepts: Embedded Structure and the principle of nominative maintenance.

2.1 Embedded Structure

Various reasons are conceivable to cause long utterances. One of them is the situation that two or more subordinate clauses or sentences are embedded into one snippet. Such Grammatical structure is called *embedded structure* (Shibatani, 1978). In an *embedded structure*, there usually exists a special sign. It is called *complement sentence indicator* and includes four kinds of linguistic expressions: "～こと[~koto]", "～の[~no]", "～と[~to]", and "～ように[~youni]". A *complement sentence indicator* indicates the end of an independent purport, which will not exert influence on the whole sentence.

A snippet which contains a *complement sentence indicator* is considered unsuitable for generating brief utterances. In other words, when we try to put a restriction on the length of a snippet, we can remove the part containing a *complement sentence indicator* from the snippet.

2.2 Nominative Maintenance

It is said that a nominative noun, or a nominative noun clause is indispensable for generating a logically and grammatically correct Japanese sentence (Shibatani, 1978). Here are two examples.

赤ちゃんはもう歩けるよ。
(The baby can already walk.)

赤ちゃんにもう歩けるよ。
(can already walk to a baby.)

The first example contains a nominative noun clause, while the second example doesn't contain a nominative noun clause, and hence is not grammatically correct.

To put it another way, if a snippet contains two or more nominative noun clauses, we will have reason to believe that the snippet might have multiple subordinate sentences. In order to obtain a shorter utterance, it is desirable to select the snippet which consists of a single sentence, or possesses a simple structure. In this way, nominative noun clause can be used as another indicator for avoiding the extraction of long snippets.

3 Utterance Generation

With the concepts stated in Section 2 in mind, we propose a method to generate utterance based on the two-starting-word Markov-chain model devised in the previous work (Nishio and Han, 2012). Section 3.1 describes the rough flow and Section 3.2 presents the specific formula to estimate the priority of each snippet candidate.

3.1 Flow of the Utterance Generation

As described in Section 2.1, one of the reasons for the emergence of long utterances is the situation that two or more subordinate clauses or sentences are embedded into one snippet. Here arises the necessity to select a snippet fragment from the snippet-candidate set extracted from Twitter.

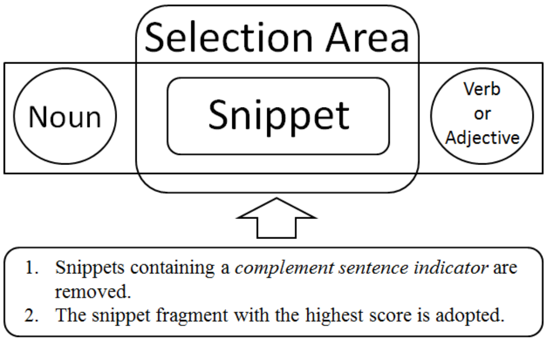
Selection Area
Noun
Snippet
Verb or Adjective
1. Snippets containing a *complement sentence indicator* are removed.
2. The snippet fragment with the highest score is adopted.

Figure 2: Processing schema of snippet selection

Figure 2 shows the overall view. Snippets containing a *complement sentence indicator* are removed from the snippet-candidate set first, then the snippet fragment with the highest priority is selected. The method to estimate the priority score for each snippet fragment will be explained in Section 3.2.

Another reason for the emergence of long utterances might have existed in the backward- or forward-direction Markov processing.

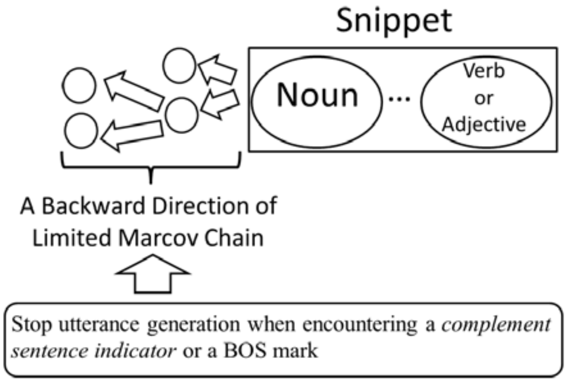
Snippet
Noun
…
Verb or Adjective
A Backward Direction of Limited Marcov Chain
Stop utterance generation when encountering a *complement sentence indicator* or a BOS mark

Figure 3: Processing schema of backward utterance generation

Backward-direction Markov processing starts with a noun, and extends to the left direction based on a bi-directional Markov dictionary as shown in Figure 3. The process will stop when encountering a *complement sentence indicator* or a BOS (Beginning of Sentence) mark.

Similarly, the forward-direction Markov processing starts with a verb or adjective, and extends to the right direction (as shown in Figure 4). The process will continue only when morphemes other than an independent word serve as a chain candidate. An independent word tends to take a pivotal role in a sentence, and is therefore likely to start a completely new statement which might lead to a long sentence finally. Moreover, if a punctuation or a EOS mark is encountered, the process will terminate at that time.

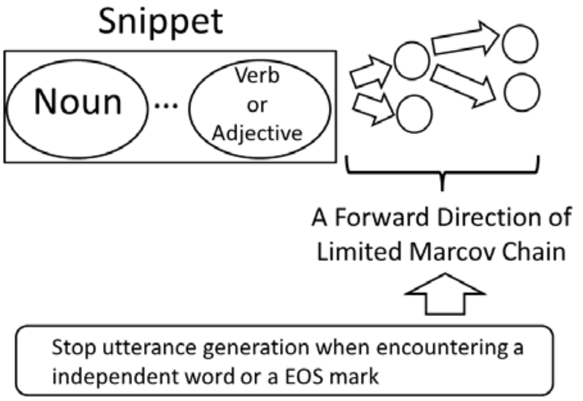
Snippet
Noun
…
Verb or Adjective
A Forward Direction of Limited Marcov Chain
Stop utterance generation when encountering a independent word or a EOS mark

Figure 4: Processing schema of forward utterance generation

3.2 Priority Estimation of Snippet Fragment

With the above concepts in mind, we define a measure below to calculate the priority score for each snippet-fragment candidate.

$$Score_i = \begin{cases} 0 & if\ (FL_i = Subst_i + Decl_i) \\ \frac{1}{M_i} & if\ ((FL_i > Subst_i + Decl_i) \& (N_i \leqq 1)) \\ \frac{1}{N_i \cdot M_i} & if\ ((FL_i > Subst_i + Decl_i) \& (N_i > 1)) \end{cases}$$

Here, $Score_i$ stands for the score assigned to each snippet fragment. FL_i indicates the total number of characters contained in the snippet fragment, and N_i indicates the number of case particles that are possible to appear with nominatives in the snippet fragment. M_i is the number of morphemes in the snippet fragment. $Subst_i$ indicates the length of the noun at the beginning of the snippet fragment, and $Decl_i$ indicates the length of the verb or adjective at the end of the snippet fragment.

FL_i is compared with the sum of $Subst_i$ and $Decl_i$ to determine whether the snippet fragment is composed only of the noun and the declinable word. Only when additional characters exist, a score other than 0 is assigned to the snippet fragment. The score varies inversely with the number of morphemes in the snippet fragment, and even gets N_i times lower when more than 2 potential nominatives might exist.

4 Emotion Presumption

In order to add more entertaining elements into the system, we implement an Emoticon Annotating function to assign an emoticon to the computer-generated utterances. Our Function consists of two steps: emotion determination and emoticon annotation. In this section, we elaborate the process to determine an emotion for a generated sentence based on machine learning techniques.

4.1 Extraction of Emotion Trigger

Our basic idea is to infer the corresponding emotion according to a particular textual clue. For example, in a sentence "突然雨が降ってきたので、残念だ" (It was regrettable that it suddenly started to rain), "突然雨が降ってきた" acts as an emotion trigger, together with the conjunction "ので" implying a causal relationship between "突然雨が降ってきたので" and "残念だ" (regrettable).

If we can find some disciplinary rules or patterns from the usage of emotion triggers, we might be able to infer the emotion even for an incomplete sentence (i.e., a sentence that doesn't contain any explicit emotion expression such as *regrettable* or *happy*.

Tokuhisa et al. have employed the combination of a conjunction and an emotion as the keyword to search the Web for emotion triggers, and performed a kNN-based similarity calculation between an input sentence and the emotion-trigger corpus to infer the emotion for the input sentence (Tokuhisa et al., 2008). In another study, Matsumoto et al. have created some sentence patterns from a small Japanese lexicon manually each with a pre-assigned emotion (Matsumoto et al., 2006). When the input sentence matches a sentence pattern, the emotion of the corresponding sentence pattern will be assigned to the input sentence.

Both works attempt to find patterns to infer emotions from input sentences. However, Tokuhisa et al. have used a very simple algorithm considering the method for pattern matching, whereas Matsumoto et al. have a major issue in the scale of data source (i.e., the small lexicon).

In this study, we combine the advantages of the above two works, and propose a new method to infer emotions based on Predicate-Argument Structure. Specifically, we collect an emotion trigger corpus from the Web in a similar way Tokuhisa et al. have done, while conduct the pattern matching using each predicate and its arguments contained in the emotion trigger corpus.

Given this perspective, the first task in our study is to extract the emotion triggers from the Web for a particular type of emotion. Here, we use Twitter as the Web data source, and search it taking the conjunction combined with the emotion expression as the clue word for emotion triggers as shown in Figure 5.

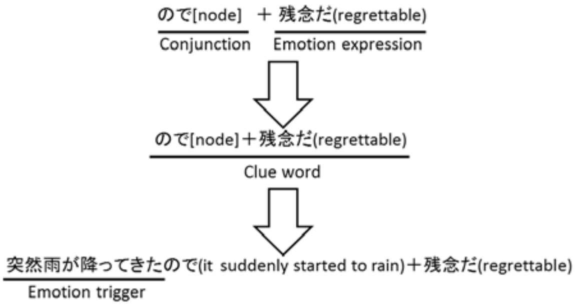
ので[node] ＋ 残念だ(regrettable)
Conjunction
Emotion expression
ので[node]＋残念だ(regrettable)
Clue word
突然雨が降ってきたので(it suddenly started to rain)＋残念だ(regrettable)
Emotion trigger

Figure 5: Flow of emotion trigger extraction

Based on the experience of Tokuhisa et al., we have used nine kinds of conjunctions including "ので" [node], "から" [kara], "ため" [tame], "のは" [nowa], "のが" [noga], "ことは" [kotowa], "ことが" [kotoga], "て" [te], and "で" [de]. Similarly, nine types of emotions and their concrete expressions are extracted from (Nakamura, 2003) and used here as emotion expressions. They include: "哀" (sorrow), "安" (ease), "厭" (hate), "喜" (hope), "驚" (surprise), "好" (like), "恥" (shame), "怒" (angry), and "怖" (fear) .

4.2 Extraction of Predicate Arguments

The emotion triggers obtained in Section 4.1 are then analyzed to extract predicate arguments using KNP [1], a free Japanese dependency analyzer. A predicate argument is the argument appearing together with a predicate.

For example, the analytical result from KNP for the sentence "私は食堂でカレーを食べた" (I ate curry at the dining hall) includes three arguments for the predicate "食べた" (ate): "私は" (I), "食堂で" (dining hall), and "カレーを" (curry). "～は" (I), "～で", and "～を" are called Ga-case, De-case, and Wo-case respectively in Japanese. Our idea is to collect the pair of a

[1] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

predicate and one of its arguments, and find the relation between a predicate argument structure and an emotion statistically. Here, we take the three-element combination (predicate, case, noun) as the basic feature in our machine learning process. For example, for the above sentence, we are able to obtain three instances for the basic feature.

(eat, Ga-case, I)
(eat, De-case, dining hall)
(eat, Wo-case, curry)

However, since the combination of declinable words, cases, and nouns could be infinite, we might need a way to abstract the basic features to avoid the data-sparseness problem. Here we use a thesaurus called Nihongo Goi Taikei (NTT Communication Science Laboratories, 1997) to accomplish this task. This thesaurus classifies all the concepts in Japanese into superordinate ones and subordinate ones in different hierarchies. For example, “flower” and “tree” are abstracted into “plant”, and “pace” and “footwork” are abstracted into “operation of hand and foot”. We generate abstracted features from the set of basic features in this way, and create a training data set containing 779,638 instances. Later in Section 6, we will talk about the different effects in using these two sorts of features.

4.3 Polarity Annotation

Abstraction is considered as a means to address the sparseness problem when generating predicate arguments. However, in case two nouns with opposite polarities have the same superordinate concept, abstraction might become kind of side effect. For example, both "矜持" (pride) and "おごり" (arrogance) are abstracted into the same superordinate concept "自信・誇り・恥・反省" (faith・glory・shame・serious), which is not desirable for subsequent machine learning. To solve this issue and make the abstraction more accurate, we annotate a polarity property to the abstracted element as shown in Figure 6. Here, we have three kinds of polarities: P, N and E, indicating Positive, Negative, and Even respectively.

It is not only during the process of abstraction that polarity is important. Suppose we have two noun phrases: "きれいな人" (a pretty person) and "失礼な人" (a rude person). The head is common to both, while the meanings are completely different because of the modifiers with opposite polarity. In most cases, we consider that the polarity of the modifier is more important than the head noun itself. In this situation, we also need to assign a polarity to the head noun, otherwise we will obtain a lot of self-contradictory feature instances, and finally impair the performance of the machine-learning based classifier.

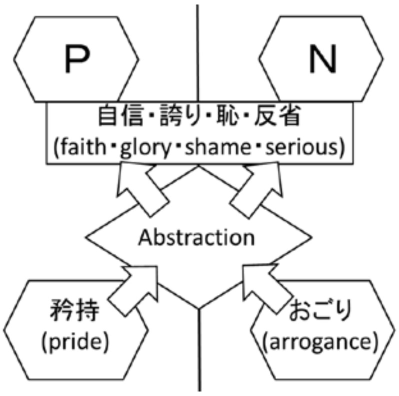

Figure 6: Example of abstraction in consideration of PN value

We annotate a polarity to the head noun according to the polarity of the modifier as shown in Table 1. Polarity information used in this study comes from two dictionaries (Kobayashi et al., 2004; Higashiyama et al., 2008; Takamura et al., 2005).

Modifier	Head noun
P	P
N	N
E	Polarity of the head noun itself

Table 1: Polarity annotation rules for modificands

4.4 Emotion Classification

Using the training data set we have created in Section 4.2 and 4.3, we employ the Naive Bayes algorithm as the basic machine learning method to generate an emotion classifier. Formula 1 shows the basic idea of Naive Bayes classifier, where *P(e)*, *P(d)*, *P(e|d)*, and *P(d|e)* indicate the probability of a emotion, the probability of a emotion trigger, the probability of an emotion provided with a particular emotion trigger, and the probability of an emotion trigger provided with a particular emotion.

$$P(e \mid d) = \frac{P(d \mid e) \times P(e)}{P(d)} \qquad (1)$$

P(e) is calculated as the ratio of the total number of the instances holding a particular emotion divided by the total number of the whole corpus. *P(d|e)* could be estimated by Formula 2.

$$
\begin{aligned}
&P(d \mid e) \\
&= P(f_1 \wedge \cdots \wedge f_j \mid e) \\
&\approx \prod_{i=1}^{j} P(f_i \mid e)
\end{aligned}
\qquad (2)
$$

$P(f_i|e)$ represents the probability of the ith feature provided with a particular emotion. We calculate *P(e|d)* for each emotion classification and identify the classification with the largest probability as the emotion for the provided emotion trigger.

5 Emoticon Annotation

In this section, we describe the procedure to annotate an emoticon to a computer-generated utterance based on the result of emotion presumption.

An emoticon could be one or more characters or symbols, or a combination of both sometimes. Generally users want to express some sort of facial expression through emoticons. Here are some examples: "＼（＾ ＾）／", " <(｀^´)> ", "o(^◇^)o".

Some previous studies have been carried out for emoticon analysis. Urabe et al. quantify the emotions expressed by emoticons through a questionnaire (Urabe, 2013). Similarly, Emura et al. create an emoticon collection and classify the emoticons according to a questionnaire (Emura, 2012).

Both works aim at providing emoticon candidates for a text input by the user. Different from the previous works, our purpose is to annotate the emoticon to a computer-generated utterance based on the emotion presumption. We adopt an emoticon database built by Kawakami in our study (Kawakami, 2008). This database contains 31 kinds of basic emoticons belonging to five emotion categories each with its own relative strength.

However, as described in Section 4, our schema has nine kinds of emotion-classification, which is different from that of Kawakami. For this reason, we have adjusted our classification to conform with the previous work as shown in Figure 7.

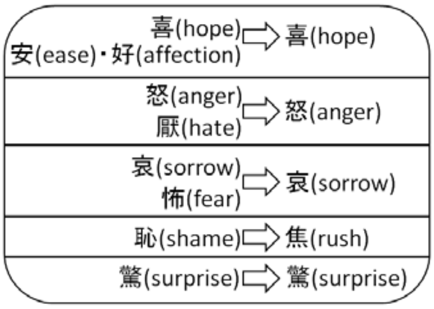

Figure 7: Emotion-classification consolidation

6 Experiments and Evaluations

We have built two kinds of prototypes based on the algorithm devised in the previous work (Han et al., 2011 & Nishio et al., 2012) and our approach respectively. Then by running each prototype constantly, we have collected a lot of execution results and conversation log data. Several evaluations are conducted to examine the effectiveness of our methods based on these results and data.

6.1 Evaluation on Utterance Generation

A subjective assessment on utterance generation is carried out with 14 college students who haven't involved in this work so far. Three conversation fragments for both prototypes are randomly extracted from the conversation log data and given to all the examinees together with some simple instructions. Then the examinees are told to compare each pair of conversation log data without being informed which log data is coming from our system in two points: Association between Utterances, and Utterance Focus. The former evaluation item indicates the association between continuous utterances, i.e., how good has the conversation topic transited? The latter item, Utterance Focus, evaluates the quality of a generated utterance sentence.

For both evaluation items, 12 students have given their votes to our system indicating that most examinees consider our system as a better solution compared with the previous work. This reveals the effectiveness of our approach to cope with the issue of long utterances occurred in the previous work, while maintaining the Two-starting-word style Markov connection and the natural transition between utterances simultaneously.

6.2 Evaluation on Emotion Presumption

Following the steps described in Section 4.1, we have extracted 779,638 emotion triggers from Twitter. The whole dataset is divided into two parts, 90% as the training data and the remaining 10% as the test data. Then we conduct a series of experiments to examine the performance of the machine-learning based emotion classifier.

According to the description in Section 4.2 and 4.3, we have employed four kinds of feature in different experiments. Table 2 shows the name of each experiment and a brief explanation on its feature. The baseline indicates the method where 2-gram model are used instead of predicate-argument structure.

Experiment	Feature
baseline	2-gram model
no_abs	not abstracted feature
abs	abstracted feature
abs_pn	abstracted feature with PN value
abs_pn’	abstracted feature with PN value from modifier

Table 2: Differences among experiments

Figure 8 shows the emotion classifying accuracies of each method varying with the volume of training data. When we use only 2,000 emotion triggers as the training data, all the methods show the poorest performance. As we increase the volume of the training data, the accuracy of each method begins to increase except the 2-gram model. Before the data volume reaches 200,000, methods using abstracted features have kept outperforming those not abstracted. This is what we have expected. When we don’t have enough training data to conduct machine learning, we will encounter the data-sparseness problem. Abstraction is expected to be an effective solution to this issue. What have been observed in Figure 8 proves the effectiveness of feature abstraction. Among the three methods involved feature abstraction, abs_pn and abs_pn’ performs better than abs, proving the usefulness of Polarity. However, there is no obvious difference between abs_pn and abs_pn during the whole process. This is what we haven’t expected and should be exhaustively investigated until the reason is clear.

On the other hand, little difference is observed when the data volume is more than 400,000, no matter the features are abstracted or not. This might indicate the turning point of data sparseness. In other words, 400,000 or more emotion triggers are likely to be sufficient for machine-learning based emotion classification.

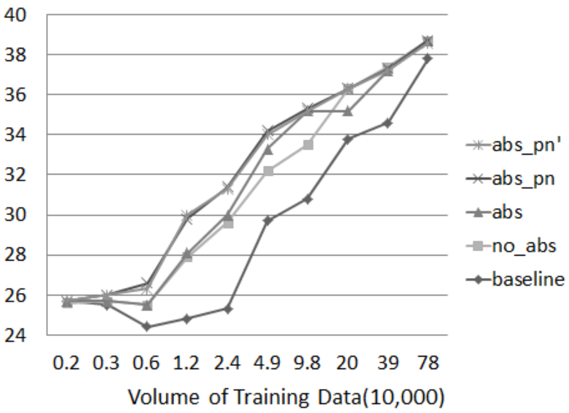

Figure 8: Change of accuracies with the volume of training data

Generally, although the classification performance is not as good as we have expected, our approach has got a better performance than the baseline method.

6.3 Evaluation on Emoticon Annotation

We randomly select five conversation fragments as the evaluation data. With the support of the examinees described in Section 6.1, we conduct another subjective assessment. The emoticon annotation function has been applied to the utterances in each fragment. Then two kinds of fragments are shown to the examinees: the original fragments and their emoticon-versions. Here are the questions in the questionnaire.

- Q.1
  Do you think conversation fragments annotated with emoticons are more interesting than those containing plain text only?
- Q.2
  For the utterances annotated with emoticons, do you think the atmosphere the emoticons are conveying conforms with the text?
  (1-yes, 2-intermediate, 3-no)

The questions inquire about the overall significance and the specific precision. Table 3 shows the average evaluation results calculated from all the examinees based on the five fragments.

Question	Average Result
Q.1	66%
Q.2	1.9

Table 3: Evaluation results on emoticon annotation

According to the evaluation results, 66% of the examinees agree that the emoticon annotation will enhance the entertaining aspect of conversation systems, and our approach seems effective to accomplish this task.

7 Conclusion

In this paper, we propose some significant improvement-strategy to a previously developed non-task-oriented conversation system.

Specifically, we introduce an automatic utterance generation in consideration of the *embedded structure* of sentences and the principle of nominative maintenance. Meanwhile, emoticon annotation based on emotion presumption has been implemented to add entertaining elements into the conversation interface with a user. The experimental results show that our approach proposed in this study has helped improve the performance of a conversation system.

However, the result is not as good as we have expected. For example, We have focused on the snippets containing nominatives while neglected the grammatically incorrect sentences during the process of sentence generation. There might exist a need to utilize the incomplete sentences too in order to increase the diversity and number of candidate snippets. Another problem lies in the emoticon annotation function. It is impossible to determine the emotion for a generated sentence if it lacks case particles according to the current method. We are going to incorporate some new strategies into the system to address these issues.

References

Emura Yuka and Seki Youhei. 2012. Facemark Recommendation based on Emotion, Communication, and Motion Type Estimation in Text. Journal of Natural Language Processing Volume 19, Number 5, 401-418. (in Japanese)

Han Dongli, Kinoshita Yasuhiro, Fukuchi Ryu, and Tsurugi Kousaki. 2011. Utterance Generation Using Twitter Replying Sentences and Character Assignment. International Journal of Digital Content Technology and its Applications, Vol. 5, No. 10, 119-126.

Han Dongli, Song Xin, and Maeda Kazuki. 2010. Topic Presentation for a Free Conversation System Based on the Web Texts. International Journal of Digital Content Technology and its Applications, vol.4, no.3, 7-14.

Higashiyama Masahiko, Inui Kentaro, and Matsumoto Yuji. 2008. Jyutugo No Sentakushikousei Ni Chakumoku Shita Meishi Hyouka Kyokusei No Kakutoku. Proceedings of the 14th Annual Conference of the Natural language Processing, 2G3-01. (in Japanese)

Higuchi Shinsuke, Rzepka Rafal, and Araki Kenji. 2008. A Casual Conversation System Using Modality and Word Associations Retrieved from the Web. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, 382-390.

Kawakami Masahiro. 2008. The Database of 31 Japanese Emoticon with their Emotions and Emphases. Osaka Shoin Women's College human science research bulletin, 7, 767-782. (in Japanese)

Kobayashi Nozomi, Inui Kentaro, Matsumoto Yuji, Tateishi Kenji, and Fukushima Toshikazu. 2004. Collecting Evaluative Expressions for Opinion Extraction. The 1st International Joint Conference on Natural Language Processing, 584-589.

Matsumoto Kazuyuki, Ren Fuji, and Kuroiwa Shingo. 2006. Emotion Estimation System Based on Emotion Occurrence Sentence Pattern. Lecture Notes in Artificial Intelligence 7114 (Computational Intelligence), Springer, 902-911.

Nakamura Akira. 2003. Kanjyo Hyougen Jiten. Toukyoudou Shuppan, Japan. (in Japanese)

Nishio Yusuke and Han Dongli. 2012. Automatic Utterance Generation by Keeping Track of the Conversation's Focus within the Utterance Window. Lecture Notes in Artificial Intelligence 7614 (Advances in Natural Language Processing), Springer, 322-332.

NTT Communication Science Laboratories. 1997. Nihongo Goi Taikei. Iwanami Shoten, Japan. (in Japanese)

Shibatani Masayoshi. 1978. Nihongo No Bunseki Seisei Bunpou No Houhou. Taishukan Shoten, japan. (in Japanese)

Song Xin, Maeda Kazuki, Kunimasa Hiroyuki, Toyota Hiroyuki, and Han Dongli. 2009. Topic Control in A Free Conver-sation System. Proceedings of the 2009 IEEE International Conference on Natural Language Processing and Knowledge Engineering, 529-534.

Takamura Hiroya, Inui Takashi, and Okumura Manabu. 2005. Extracting Semantic Orientations of Words using Spin Model. Proceedings of the 43rd Annual Meeting of the ACL. 133–140.

Tokuhisa Ryoko, Inui Kentaro, and Matsumoto Yuji. 2008. Emotion Classification Using Massive Examples Extracted from the Web. Proceedings of the 22nd International Conference on Computational Linguistics (COLING-2008), 881-888.

Urabe Yuki, Rzepka Rafal, and Araki Kenji. 2013. Kaomoji No Arawasu Kanjyou Wo Motiita Kaomoji Suisen Syetem No Koutiku. Proceedings of the 19th Annual Conference of the Natural 1language Processing. P3-20, 648-651. (in Japanese)