Building a Conversational Model from Two-Tweets

Ryuichiro Higashinaka[1], Noriaki Kawamae[2], Kugatsu Sadamitsu[1], Yasuhiro Minami[3]
Toyomi Meguro[3], Kohji Dohsaka[3], and Hirohito Inagaki[1]

[1] *NTT Cyber Space Laboratories, NTT Corporation*
1-1 Hikarinooka, Yokosuka, Kanagawa, 239-0847 Japan
{higashinaka.ryuichiro,sadamitsu.kugatsu,inagaki.hirohito}@lab.ntt.co.jp

[2] *NTT Comware Corporation*
1-6 Nakase, Mihama-ku, Chiba 261-0023 Japan
kawamae.noriaki@nttcom.co.jp

[3] *NTT Communication Science Laboratories, NTT Corporation*
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
{minami.yasuhiro,meguro.toyomi,dohsaka.kohji}@lab.ntt.co.jp

***Abstract*—The current problem in building a conversational model from Twitter data is the scarcity of long conversations. According to our statistics, more than 90% of conversations in Twitter are composed of just two tweets. Previous work has utilized only conversations lasting longer than three tweets for dialogue modeling so that more than a single interaction can be successfully modeled. This paper verifies, by experiment, that two-tweet exchanges alone can lead to conversational models that are comparable to those made from longer-tweet conversations. This finding leverages the value of Twitter as a dialogue corpus and opens the possibility of better conversational modeling using Twitter data.**

I. Introduction

Twitter offers useful data for analyzing social communication, such as how information spreads among people [1], [2]. Twitter data can also be useful for mining timely opinions on the web for marketing purposes [3]. Recently, because of the conversational nature of Twitter (e.g., mentions and replies) and because of its vast data size and diversity of content compared to what the dialogue research community has seen, there has been emerging work on creating stochastic conversational models, such as hidden Markov models (HMMs), from Twitter data [4]. Building such models can be useful for analyzing how humans exchange utterances and obtaining insight into building automated text/spoken dialogue systems.

Although it is true that Twitter data are conversational and large in size, there is a severe limitation when we want to build a conversational model from Twitter data. That is, only a very small proportion of all posts form conversations, and, what is worse, most of such conversations consist of only two tweets (a post and a reply; hereafter referred to as **two-tweets**). According to a previous study, 37% of English tweets are reported to be conversational [5], of which 69% are two-tweets [4]. This tendency is the same in Japanese. According to our study (see Section IV-A), only about 5.1% of all Japanese tweets form conversations, of which 91% are two-tweets. Note that we call two or more tweets connected with an in-reply-to relationship a conversation, which is different from Kelly's idea. He regards all tweets starting from @ (an indicator of addressing other users) conversational, which we consider to be too lenient since there may not be a reply to an address.

Typically, when we want to build a conversational model, we need conversations longer than just two tweets, because conversations in general are not simply about a single interaction, such as a question and answer. Minimally, we need conversations of three or more tweets (hereafter referred to as **long-tweets**) to build reasonable conversational models when we want to deal with more-than-one-shot interactions. This is why previous studies have used conversations longer than three tweets to train their models [4]. Now the problem is that we have a very small number of long-tweets, which severely limits the size of training data for conversational modeling.

This paper proposes to make use of two-tweets to build a conversational model that can be equivalent to or better than one built from long-tweets. Our basic idea is to cluster tweets within two-tweets to form pseudo long conversations, from which we can train a conversational model. In our approach, we use a non-parametric Bayesian method called the infinite HMM for the modeling. Our approach could make full use of the conversations in Twitter and has the potential to leverage the performance of conversational models that can be learned from Twitter data.

II. Related Work

Modeling conversations from dialogue data has long been studied, mainly by using HMMs for their usefulness in dealing with sequential data. Shirai trained ergodic HMMs from the data of task-oriented spoken dialogues to analyze the functions of audio-visual information in dialogue [6]. Dialogues with less task restriction have also been modeled using HMMs; Isomura et al. used HMMs to model interview-like conversations and used the HMMs to evaluate the naturalness of conversations [7], and Meguro et al. used HMMs to analyze counseling-like listening-oriented dialogues for the purpose of building listening agents [8]. Engelbrecht et al., and Higashinaka et al., both trained HMMs that can predict turn-wise user satisfaction transitions within a dialogue [9], [10]. All these studies use long conversations that at least last

three turns; typically one conversation lasts more than a few minutes, resulting in tens of turns per dialogue. The problem in these studies is that collecting conversations using human subjects is usually very costly, resulting in a limited number of dialogues with restrictions in dialogue topics.

Twitter data could overcome such a problem by its data size and diversity in content, but as we mentioned in the introduction, modeling Twitter conversations needs to deal with the scarcity of long conversations. Currently, only long-tweets are used for training conversational models as in [4], which we consider does not make the best use of conversations in Twitter. To the best of our knowledge, no work has tackled the problem of the scarcity of long conversations in Twitter.

In the context of modeling Twitter data, Ramage et al., used a technique called labeled latent Dirichlet allocation (labeled LDA) to model a large number of tweets in order to understand the varied content in Twitter [11], but their focus is to cluster individual tweets by latent topics, not to model conversations. Joty et al. proposed to model conversations using an HMM (called HMM+Mix) [12]; however, they use the data of e-mail discussions and messages at Internet forums, where the conversations are typically longer than those found in Twitter, and therefore do not address the lack of long conversations.

III. Approach

We aim to use two-tweets to create conversational models that have only been realized by using long-tweets. We consider that, by using clustering techniques, we can cluster tweets within two-tweets to form pseudo long conversations, from which we can train a conversational model for long conversations; that is, we first find two-tweets, such as $A \rightarrow B$ and $B' \rightarrow C$, where B and B' are similar tweets, and cluster the similar ones to form a pseudo three-tweet conversation (i.e., $A \rightarrow \{B, B'\} \rightarrow C$). If we can apply this process to many two-tweets and obtain the optimal clusters with their transitions, then that structure would make a model for long conversations.

Since this process is the same as what HMM training does, we can adopt known algorithms of HMMs for this task. In this paper, we employ a non-parametric Bayesian version of the HMM; namely, the infinite HMM, and use Gibbs sampling for parameter estimation. The infinite HMM is related to the Dirichlet process [13], which is a non-parametric Bayesian model [14]. The hierarchical Dirichlet processes (HDP) [15] is one realization of the Dirichlet process for handling mixture models, and the infinite HMM is one implementation of the HDP. The infinite HMM has been applied to modeling sequential data when the number of states is not known in advance; the optimal number of states is determined by data. We find this feature useful because Twitter data are diverse in content and it is difficult to estimate the number of states in advance. In addition, a Bayesian approach has been reported to perform better [4] than the EM algorithm [16].

One possible drawback of the above approach is that Gibbs sampling is computationally heavy. Even when we are to use only two-tweets in Twitter, which constitute a small proportion of all tweets, we still need to handle a large number of tweets. Since we have more than a million two-tweets in our data set (see Section IV-A), a straightforward application of Gibbs sampling would be extremely difficult. Therefore, in our approach, we insert one step before HMM training; that is, the creation of small subsets. To make such subsets, we take a grep approach; we grep the entire tweets by keywords to create keyword-related subsets of tweets. Compared to applying sophisticated hard-clustering algorithms for creating subsets of data, we find this approach promising and attractive because we can ascertain that a subset is concerned with a certain topic semantically constrained by keywords. One difficulty of this approach is that tweets are generally short, and this simple grep approach may end in low recall. To overcome this problem, following [17], we turn to Wikipedia to annotate tweets with Wikipedia concepts (titles/entries) so that we can additionally include such concepts as our grep target and thereby leverage the coverage of tweet extraction.

We first describe how we use Wikipedia to annotate tweets with Wikipedia concepts in order to create subsets. Then, we describe how we train our model using the infinite HMM.

A. Making a Subset of Tweets using Wikipedia

Banerjee et al. annotated tweets with Wikipedia concepts [17]. They first made a text database of Wikipedia articles. Then, for each tweet, they queried the database using the words in the tweet to retrieve top-N Wikipedia articles (they used 20 for N). They used the titles of the retrieved articles to semantically augment tweets for better clustering.

Our approach is similar to theirs in that we use Wikipedia to complement the information of tweets, but different in that we take a more direct approach. We first add all Wikipedia titles to the dictionary of a morphological analyzer so that words that match Wikipedia titles can be detected in morphological analysis. Second, we create a database of Wikipedia titles and their categories (NB. Wikipedia usually has several categories associated with each title). Since Wikipedia categories have a hierarchical structure, for each title, we also include categories that are one layer above the categories directly associated with that title. Such hypernym categories can be useful for adding generalized meanings to the titles. For example, the word "w-cup (*daburyuu hai*)" is associated with a category "world cup (*wârudo kappu*)" together with their upper layer categories "world championships (*sekai senshuken*)" and "international sports competitions (*kokusai supôtsu kyougi taikai*)". Third, we process all tweets by the morphological analyzer and detect Wikipedia titles. The detected titles are then coupled with their categories using the database. Finally, given such processed tweets and keywords KW, we can create a subset of the tweets by finding those that contain KW. Note that grep extracts tweets that have KW in the tweets as well as in the associated Wikipedia categories. In this approach, for example, a tweet having "w-cup" can be extracted by keywords such as "world-cup" as well as "sports" and "championships".

B. Modeling Tweets by the Infinite HMM

In the infinite HMM, tweets are processed one by one. The first tweet is clustered to the initial cluster (NB. there is only one cluster at the beginning). Then, the next tweet t_i is clustered to an already occupied cluster c_j or creates a new cluster ($c_{j=new}$) with the probability

$$\mathrm{P}(c_j|t_i) \propto \mathrm{P}(c_j|c_{t_{i-1}}) \cdot \mathrm{P}(c_{t_{i+1}}|c_j) \cdot \mathrm{P}(t_i|c_j),$$

where c_t means the cluster of a tweet t. In a conversation, tweets are given a sequential order; t_{i-1} and t_{i+1} denote the previous and next tweets of t_i. $\mathrm{P}(c_k|c_j)$ is a transition probability:

$$\mathrm{P}(c_k|c_j) = \frac{\mathrm{transitions}(c_j, c_k) + \beta}{\sum_{l=1}^{K} \mathrm{transitions}(c_j, c_l) + K \cdot \beta + \alpha},$$

where α is the hyper-parameter that determines how likely a new cluster is created, K is the number of occupied clusters, and $\mathrm{transitions}(c_j, c_k)$ returns the number of transitions from c_j to c_k. β is a flooring value to avoid zero probability. $\mathrm{P}(t_i|c_j)$ is the probability that t_i is generated from c_j; that is,

$$\mathrm{P}(t_i|c_j) = \prod_{w \in W} \mathrm{P}(w|c_j)^{\mathrm{count}(t_i, w)},$$

$$\mathrm{P}(w|c_j) = \frac{\mathrm{count}(c_j, w) + \gamma}{\sum_{w \in W} \mathrm{count}(c_j, w) + |W| \cdot \gamma},$$

where W is a set of features (e.g., bag-of-words), $\mathrm{count}(*, w)$ a function that returns the number of occurrences of a feature w for a tweet or a cluster, and γ the hyper-parameter.

The probability of creating a new cluster is

$$\mathrm{P}(c_{new}|c_{t_{i-1}}) \cdot \mathrm{P}(c_{t_{i+1}}|c_{new}) \cdot \mathrm{P}(t_i|c_{new}),$$

where $\mathrm{P}(c_{new}|c_{t_{i-1}})$ and $\mathrm{P}(c_{t_{i+1}}|c_{new})$ are derived by

$$\mathrm{P}(c_{new}|c_{t_{i-1}}) = \frac{\alpha}{\sum_{l=1}^{K} \mathrm{transitions}(c_{t_{i-1}}, c_l) + \alpha},$$

$$\mathrm{P}(c_{t_{i+1}}|c_{new}) = \frac{1}{K+1}.$$

Here, we use a uniform distribution for $\mathrm{P}(t_i|c_{new})$.

After all tweets have been processed, Gibbs sampling is performed; that is, we repeatedly select one of the tweets from its cluster and relocate that tweet as if it were the last tweet to be clustered. After performing a sufficient number of samplings, we obtain the optimal number of clusters together with their emission probabilities and transition probabilities, which become our conversational model.

Determining the appropriate features for representing tweets is a difficult problem. Following previous studies [4], [12], we use bag-of-word-unigrams in this paper. This choice of features is also backed by the fact that there are too many unique words in Twitter data (see Section IV-A), suggesting that features made using bigrams or longer n-grams would be too sparse.

TABLE I
STATISTICS OF OUR TWITTER CORPUS.

	food	sports	all
# conversations	63312	37292	1211725
# tweets	132203	78123	2500918
# words	2517179	1870382	40098705
# unique words	74865	75309	452099

IV. EXPERIMENT

To verify our approach, we performed an experiment. We first collected tweets from the public timeline. Then, we made two subsets of tweets related to food and sports, which we consider to be common topics in everyday conversation. Since our aim is to examine whether the two-tweets can be used to create models that can be achieved by long-tweets, we further divided the subsets into two-tweets and long-tweets. Finally, we examined whether conversational models made from two-tweets can compete with those made from long-tweets.

A. Data Collection

We collected Japanese tweets from the public timeline using Rest and Streaming APIs between February 2 and September 15, 2010. We used the default access level called "Spritzer". In this period, we crawled over 95 million tweets (95,501,894 tweets). From them, we retrieved conversations using the in-reply-to field of the tweets (4,907,519 tweets; 5.1%). After discarding conversations starting with replies, the resulting corpus of conversations contains about 2.5 million tweets. This is about 2.62% of the crawled data, which shows just how scarce conversations are.

Having created the corpus, we ran our morphological analyzer JTAG [18] with a user defined dictionary augmented with Wikipedia titles (see Section III-A). We used the Japanese Wikipedia dump of April 20, 2011. We also created a database of titles and their categories from the same dump. After all tweets had been annotated with Wikipedia categories, we created two subsets by grep using "meal|food (*shokuji*|*ryouri*)" ('|' indicates OR) and "sports (*supôtsu*)" as keywords. We call the subsets here Food-Set and Sports-Set. Here, grep was done on the conversation level; that is, when one of the tweets in a conversation had a keyword, that entire conversation was extracted.

Table I shows the statistics of our Twitter corpus together with the information of the two subsets. We can confirm the diversity of content in Twitter from its large vocabulary size. From the fair size of the subsets, we can also confirm the effectiveness of our utilizing Wikipedia. Table II shows the number of N-tweet conversations. We can see that most are two-tweets. We can also see that, as N increases, the number of tweets decreases exponentially. This shows the difficulty of using Twitter data as a dialogue corpus and makes clear the need to make use of two-tweets, which account for over 90% of all conversations (2,280,402 tweets; 91% of all tweets in our conversations).

B. Evaluation Procedure

We divided each subset into two sets. One set comprises only two-tweets (*the 2-tweet set*) and the other set long-tweets

TABLE II
NUMBER OF N-TWEET CONVERSATIONS.

N	food	sports	all
2	58269 (92.03%)	34114 (91.48%)	1140201 (94.10%)
3	4565 (7.21%)	2865 (7.68%)	66223 (5.47%)
4	426 (0.67%)	273 (0.73%)	4729 (0.39%)
5	46 (0.07%)	34 (0.09%)	506 (0.04%)
6	6 (0.01%)	5 (0.01%)	62 (0.01%)
7	0 (0.00%)	0 (0.00%)	3 (0.00%)
8	0 (0.00%)	1 (0.00%)	1 (0.00%)
3 $\leq$	5043 (7.97%)	3178 (8.52%)	71524 (5.90%)

(*the long-tweet set*). We further divided the long-tweet set into two sets by dividing it in half; we call them *the long-tweet train set* and *the long-tweet test set*. In Food-Set, the long-tweet train set and long-tweet test set have 2522 and 2521 conversations, respectively. In Sports-Set, the long-tweet train set and long-tweet test set both have 1589 conversations.

Since we are interested in how the conversational model made from two-tweets competes with that made from long-tweets, we trained models from the 2-tweet set and the long-tweet train set and evaluated them by looking at how they explain unseen long conversations; that is, the long-tweet test set. We also used the long-tweet test set to train a model and evaluate it using itself as test data (i.e., closed test), which will indicate the upper bound. We are also interested in how a model improves when we increase the training data. To investigate this, we split the 2-tweet set into blocks of 1000 conversations each, and investigated how a model improves by adding blocks to the training data.

C. Evaluation Metrics

For evaluation metrics, we used log likelihood (LL) and Kendall's tau, which have been used in a previous study [4]. For LL, we used the forward algorithm for calculation [16]. For Kendall's tau, we first created all possible orders of tweets in each conversation in the long-tweet test set, and calculated the LL of each order and selected the order with the highest LL. Then, that order was compared against the original order to calculate Kendall's tau by:

$$\mathrm{tau}(R, H) = \frac{n_+(R, H) - n_-(R, H)}{\mathrm{combination}(R)},$$

where R and H denote reference and hypothesis orders, $n_+(R, H)$ the number of correct pairwise orders, $n_-(R, H)$ the number of incorrect pairwise orders, and $\mathrm{combination}(R)$ the number of possible pairwise orders. The high value in Kendall's tau suggests that the flow of conversation has been successfully modeled.

D. Training infinite HMMs

We trained our infinite HMMs from (a) the 2-tweet set, (b) the long-tweet train set, and (c) the long-tweet test set of Food-Set and Sports-Set. We call the models trained from (a), (b), and (c) *the 2-tweet model*, *the long-tweet open model*, and *the long-tweet closed model*, respectively. We used tentative values of 0.01 for all α, β, and γ in HMM training (cf. Section III-B). For features, we used bag-of-word-unigram features, where the words are the top-5000 words in the 2-tweet set. The number of top-N words follows the convention [4], [12]. The number of iterations for Gibbs sampling was set to 1000, which means that each tweet is considered 1000 times for relocation.

TABLE III
THE NEGATIVE LOG LIKELIHOOD (LL) AND KENDALL'S TAU FOR THE LONG-TWEET TEST SET BY THE 2-TWEET MODEL, THE LONG-TWEET OPEN MODEL, AND THE LONG-TWEET CLOSED MODEL. * AND + INDICATE STATISTICAL SIGNIFICANCE ($p < 0.01$) OVER THE 2-TWEET AND LONG-TWEET OPEN MODELS, RESPECTIVELY.

	2-tweet	long-tweet open	long-tweet closed
Food-Set LL	290.70^{+}	294.71	285.54^{*+}
Food-Set tau	0.312^{+}	0.247	0.277
Sports-Set LL	332.36	331.92^{*}	320.86^{*+}
Sports-Set tau	0.308^{+}	0.170	0.303^{+}

E. Results

Table III shows the negative log likelihood (the lower, the better) and Kendall's tau averaged over all test conversations (i.e., the long-tweet test set) for the obtained models. We performed the Wilcoxon rank-sum test to check whether the models are significantly different. As a result, we found that the 2-tweet models significantly outperform the long-tweet open models in almost all cases and that their performance can even attain the level of the long-tweet closed models in some cases; for example, there is no statistical difference between the 2-tweet model and the long-tweet closed model for tau in both Food-Set and Sports-Set. This indicates that it is possible to use two-tweets to model long conversations.

Figures 1 and 2 show the line plots of the negative LL and Kendall's tau against the long-tweet test set depending on the size of the 2-tweet set of Food-Set and Sports-Set, respectively. The solid blue and red lines indicate the possible upper bounds (i.e., results of the long-tweet closed models) of the LL and tau, respectively. The dotted blue and red lines indicate the performance of the long-tweet open models. As for the LL, as we increase the training data, the performance gradually passed or closely neared the dotted line. Since the performance reaches that of the long-tweet open models when the number of training conversations is 5000-10000, this number of two-tweets is what we need to compete against 1500-2500 (i.e., the size of our long-tweet train sets) long-tweets. It is also noticeable that the LL somewhat degrades after 20000 two-tweets, probably because of some structural changes in the HMMs to better cope with longer tweets, which instead led to improvements in tau. The result for Kendall's tau seems better than that for the LL; the 2-tweet models steadily reached the performance of the long-tweet closed models. This indicates that, in terms of ordering, two-tweets seem very helpful for creating better conversational models.

Figure 3 shows the number of states of our 2-tweet models depending on the size of training data. Remember that the optimal number of states is automatically decided from data in the infinite HMM. It can be seen that we require about 35-40 states for the food and sports topics. This number is close to that reported in [4] when their models' performance saturated. Although we need further verification by using different hyper-parameters, 35-40 could be the rough estimate of utterance

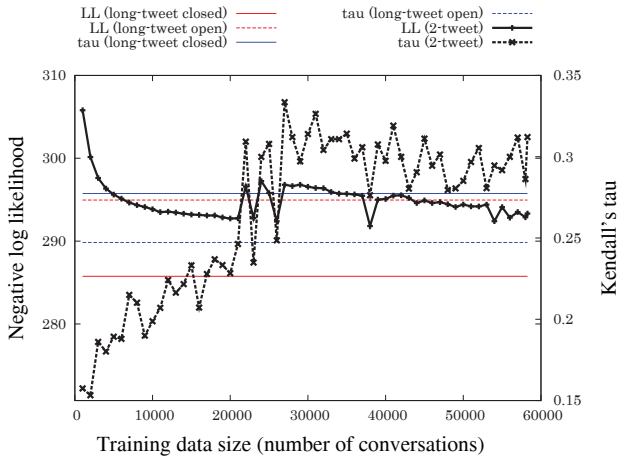

Fig. 1. Learning curve: the negative log likelihood (LL) and Kendall's tau against the long-tweet test set depending on the size of training data (2-tweets) for Food-Set. The red solid and dotted lines indicate the LL for the long-tweet closed and open models, respectively. The blue solid and dotted lines indicate Kendall's tau for the long-tweet closed and open models, respectively.

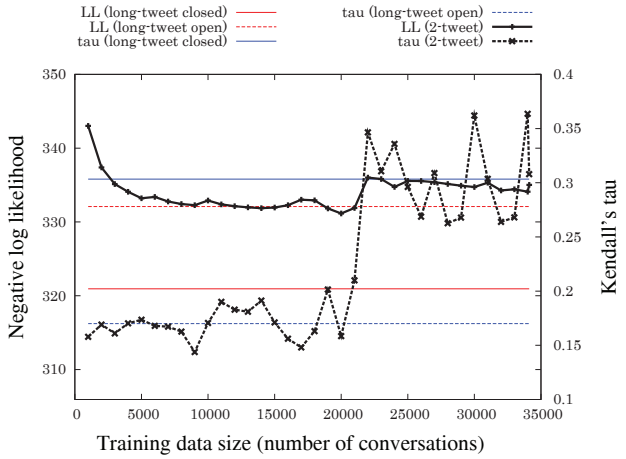

Fig. 2. Learning curve: the negative log likelihood (LL) and Kendall's tau against the long-tweet test set depending on the size of training data (2-tweets) for Sports-Set. See Fig. 1 for the descriptions of the red and blue solid and dotted lines.

variations in Twitter.

F. Analysis of Obtained HMMs

To understand our obtained HMMs, we looked at how our 2-tweet models decode long-tweets. For this purpose, we obtained best paths for the long-tweet test set in the 2-tweet model using the Viterbi algorithm. Then, we visualized frequent paths in the best paths. Figure 4 shows the resulting graph. Here, only the paths that occurred more than 15 times are shown. States that do not have such frequent paths and the paths to the END state have been removed for better visibility. The numerical numbers along the edges indicate the probabilities of choosing those paths. The graph clearly shows the existence of long sequences of tweets in the trained model even though the model is trained from two-tweets, confirming the feasibility of using two-tweets to model long conversations.

To further analyze the graph, we examined what words are

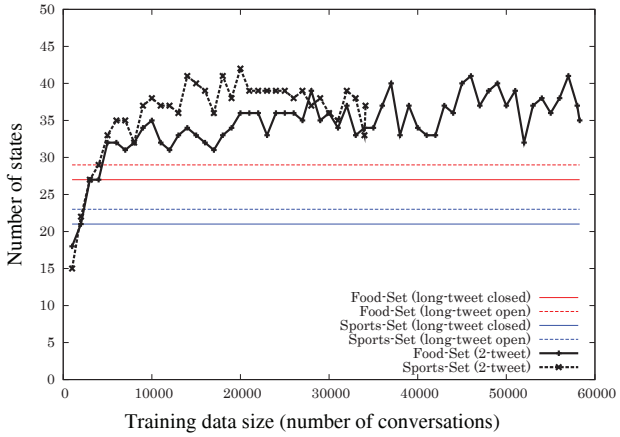

Fig. 3. The number of optimal states depending on the training size.

representative in the states. Table IV shows the representative words for some of the states in Fig. 4. To select the representative words, we used the log-likelihood ratio test, which is similar to the chi-square test; that is, for a state in question, we count the occurrences of a word in the tweets clustered to that state (see Section III-B) and tested whether that count is significantly higher than its expected value. Since the log-likelihood ratio follows the chi-square distribution, we selected representative words that had the log-likelihood ratio of over 15.13 ($p<0.0001$).

To make it easier to grasp the underlying meaning of the representative words in the table, we included our interpretations of them in square brackets. For example, in state 29, we see words that concern the status of people; e.g., come home, wake up, work, and sleepy. State 11 also concerns one's status, but is more oriented towards activities at home. In state 13, we have interrogatives. In state 6, 18, and 31, we have some response variations; namely social, affection, and emotion. In state 26, we see words that report one's meals, and in state 27, we see the description of food, such as food names and ingredients. States 26 and 27 seem to be somewhat topic-dependent states. State 7 seems to represent a generic comment; people expressing their opinions/attitudes in reply to previous tweets. Because many states come in to this state and few edges go out, this state seems to be the terminal of conversation in Twitter.

Below, we list some sequences from the graph. The sequences seem reasonable and bears some similarity to the typical conversation flow in Twitter [4], again showing the feasibility of using two-tweets for conversational modeling.

- 11:status (home) → 6:response (social) → 7:comment
- 29:status → 31:response (emotion) → 18:response (affection) → 7:comment
- 26:report → 13:question → 27:description → 7:comment

V. Summary and Future Work

In this paper, we first pointed out that Twitter data scarcely contain long conversations and that most of the conversations (91% in Japanese tweets) are composed of two tweets.

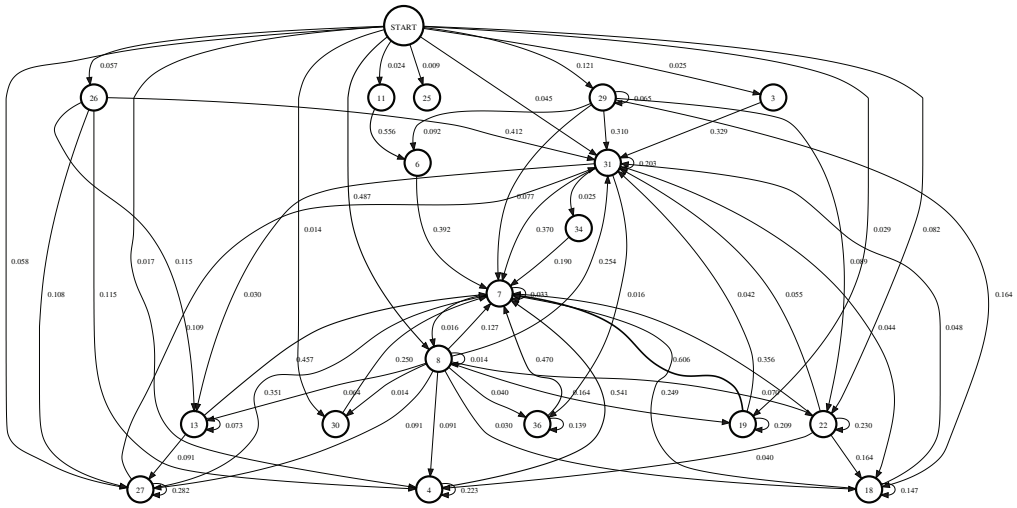

Fig. 4. Graphical representation of transitions between states for the food topic.

TABLE IV
REPRESENTATIVE WORDS OF EACH STATE FOR THE FOOD TOPIC. ID MEANS THE STATE ID. THE SQUARE BRACKETS INDICATE OUR INTERPRETATIONS OF THE REPRESENTATIVE WORDS. ENGLISH WORDS ARE TRANSLATIONS BY THE AUTHORS AND THE ORIGINAL JAPANESE WORDS ARE GIVEN IN PARENTHESES.

ID	[Interpretation] Representative words
6	**[response (social)]** welcome home (*kaeri, kae, nasai, nasa*), you must be tired (*otsukaresama*), good night (*oyasumi*)
7	**[comment]** first person pronoun (*watashi*), sentence-ending particles that express confirmation or agreement (*ne, yo, yone, desune*), think (*omou*), lol (*warai*)
11	**[status (home)]** come home (*kitaku*), I'm home (*tadaima*), !, meal (*gohan*), now (*nau, ima*), from now (*korekara*), bath (*furo*), dinner (*yuuhan*), finished (*shuuryou*)
13	**[question]** what (*nani*), okay (*daijoubu*), ?, where (*doko*), what kind of (*donnna*), how (*dou*)
18	**[response (affection)]** please (*kudasai*), !, thank you (*arigatou*), congratulations (*omedetou*), good luck (*ganbaru*), I hope to (*yoroshiku*), take care (*daiji*)
26	**[report]** today (*kyou*), lunch (*ohiru*), tonight (*konya*), yummy (*umauma*), ramen noodles (*râmen*), hamburger (*hanbâgu*), curry (*karê*), set meal (*teishoku*)
27	**[description]** salad (*sarada*), tomato (*tomato*), fried (*itame*), miso soup (*miso shiru*), soy sause (*shouyu*), vegetables (*yasai*), pork (*butaniku*), onion (*tamanegi*)
29	**[status]** meal (*gohan*), come home (*kitaku*), sleep (*neru*), wake up (*okiru*), work (*shigoto*), sleepy (*nemui*)
31	**[response (emotion)]**), (, *, ^, ·, ‘, ∀, o, ;, -, ▽, ’, ≤, ≥, !, _ (characters used for facial expressions)

Previous work has utilized only conversations lasting longer than three tweets for dialogue modeling so that more than a single interaction can be successfully modeled. This paper has verified by experiments that two-tweets alone can also lead to good conversational models that are comparable to those made from long-tweets. This finding leverages the value of Twitter as a dialogue corpus and points the way to making better use of conversations in Twitter for conversational modeling. As future work, we want to consider the possibility of using single tweets to enhance our models. Since our approach is analogous to estimating trigrams from bigrams, the same can be said for unigrams. Although single tweets do not affect the transitions, we may be able to obtain better emission probabilities. We also wish to build an automated dialogue system based on the obtained conversational models.

REFERENCES

[1] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, “Everyone’s an influencer: quantifying influence on Twitter,” in *Proc. WSDM*, 2011, pp. 65–74.

[2] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts, “Who says what to whom on Twitter,” in *Proc. WWW*, 2011, pp. 705–714.

[3] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, “Target-dependent Twitter sentiment classification,” in *Proc. ACL-HLT*, 2011, pp. 151–160.

[4] A. Ritter, C. Cherry, and B. Dolan, “Unsupervised modeling of Twitter conversations,” in *Proc. NAACL-HLT*, 2010, pp. 172–180.

[5] R. Kelly, “Pear analytics Twitter study (whitepaper),” 2009.

[6] K. Shirai, “Modeling of spoken dialogue with and without visual information,” in *Proc. ICSLP*, vol. 1, 1996, pp. 188–191.

[7] N. Isomura, F. Toriumi, and K. Ishii, “Evaluation method of non-task-oriented dialogue system using HMM,” *IEICE Transactions on Information and Systems*, vol. J92-D, no. 4, pp. 542–551, 2009.

[8] T. Meguro, R. Higashinaka, K. Dohsaka, Y. Minami, and H. Isozaki, “Analysis of listening-oriented dialogue for building listening agents,” in *Proc. SIGDIAL*, 2009, pp. 124–127.

[9] K.-P. Engelbrecht, F. Gödde, F. Hartard, H. Ketabdar, and S. Möller, “Modeling user satisfaction with hidden Markov models,” in *Proc. SIGDIAL*, 2009, pp. 170–177.

[10] R. Higashinaka, Y. Minami, K. Dohsaka, and T. Meguro, “Issues in predicting user satisfaction transitions in dialogues: Individual differences, evaluation criteria, and prediction models,” in *Proc. IWSDS*, 2010, pp. 48–60.

[11] D. Ramage, S. Dumais, and D. Liebling, “Characterizing microblogs with topic models,” in *Proc. ICWSM*, 2010.

[12] S. Joty, G. Carenini, and C.-Y. Lin, “Unsupervised approaches for dialog act modeling of asynchronous conversations,” in *Proc. IJCAI*, 2011.

[13] T. S. Ferguson, “A Bayesian analysis of some nonparametric problems,” *Annals of Statistics*, vol. 1, pp. 209–230, 1973.

[14] J. K. Ghosh, *Bayesian Nonparametrics*. Springer, 2003.

[15] Y. Teh, M. Jordan, M. Beal, and D. Blei, “Sharing clusters among related groups: Hierarchical Dirichlet processes,” in *Proc. NIPS*, 2004.

[16] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Readings in speech recognition*, vol. 53, no. 3, pp. 267–296, 1990.

[17] S. Banerjee, K. Ramanathan, and A. Gupta, “Clustering short texts using Wikipedia,” in *Proc. SIGIR*, 2007, pp. 787–788.

[18] T. Fuchi and S. Takagi, “Japanese morphological analyzer using word co-occurrence—JTAG,” in *Proc. COLING-ACL*, vol. 1, 1998, pp. 409–413.