CAAI Transactions on Intelligence Technology

Research Article

Slang feature extraction by analysing topic change on social media

ISSN 2468-2322
Received on 19th November 2018
Revised on 7th January 2019
Accepted on 11th January 2019
doi: 10.1049/trit.2018.1060
www.ietdl.org

Kazuyuki Matsumoto ✉, *Fuji Ren*, *Masaya Matsuoka*, *Minoru Yoshida*, *Kenji Kita*

Graduate School of Technology, Industrial and Social Sciences, Tokushima University, 770-8506, Tokushima-shi, Minamijosanjima-cho 2-1, Japan
✉ *E-mail: matumoto@is.tokushima-u.ac.jp*

Abstract: Recently, the authors often see words such as youth slang, neologism and Internet slang on social networking sites (SNSs) that are not registered on dictionaries. Since the documents posted to SNSs include a lot of fresh information, they are thought to be useful for collecting information. It is important to analyse these words (hereinafter referred to as 'slang') and capture their features for the improvement of the accuracy of automatic information collection. This study aims to analyse what features can be observed in slang by focusing on the topic. They construct topic models from document groups including target slang on Twitter by latent Dirichlet allocation. With the models, they chronologically the analyse change of topics during a certain period of time to find out the difference in the features between slang and general words. Then, they propose a slang classification method based on the change of features.

1 Introduction

With the expansion of social networking sites (SNSs) such as Twitter, the Internet has penetrated into daily life. The documents posted on SNSs by many users should be useful for collecting timely information. However, the documents on SNSs often include youth slang, neologism and Internet slang that are not registered into dictionaries. Therefore, analysis of slang is important for allowing extraction of new information with high accuracy.

This study focuses on the knowledge that the usage of some slang words varies over time. Therefore, we analyse a set of tweets including slang with latent Dirichlet allocation (LDA) and output their topics monthly.

Then, the randomly selected topics are set as a basis, and the similarities between the topics are calculated for each month. We investigate the topics where the similarities are greatly changed from the basis so that we can examine the characteristics of the slang based on the change of topics. We also propose a method to judge the words, whether they are used as slang, from the obtained topic change features.

This paper comprises six sections. Section 2 describes the related works of slang analysis or slang processing. Section 3 explains our proposed method to analyse slang features. In Section 4, the experiments and results are illustrated. Section 5 discusses the analysis result and the problems of the proposed method. Finally, Section 6 concludes the paper.

2 Related works

There are previous studies focused on topic detection, topic tracking or detection of slang. As follows, the related works are explained in sequence.

2.1 Topic detection

Recently, latent topic analysis has focused on the field of text/web mining or natural language processing. LDA [1] is a famous algorithm for topic modelling.

Studies on what kinds of topics appear in a document are included in the study field of topic detection. There are studies for electronic text data, such as web data [2–13]. Although there are various types of documents such as news articles, e-mails, weblogs, among others, document classification topics are often annotated with correct labels.

Kimura *et al.* [14] proposed the classification method of the relation between hashtags by considering co-occurrence and latent topic from tweets. Their method can identify the meaning of unknown hashtag by classifying relation with known hashtags. They defined the relation of hashtags as four types. As an evaluation experiment, the proposed method based on co-occurrence and the latent topic could achieve higher accuracy than the baseline methods which use only co-occurrence or latent topic.

However, there might be more than one topic for each document, and unexpected topics actually often appear. Therefore, LDA has been used more recently as a method that does not define topics in advance.

Hashimoto *et al.* [15] proposed a topic detection method by using a neural network-based vector space model to calculate semantic similarities between system review documents. Their method can achieve a higher sensitivity of eligible studies and a reduced manual annotation cost than the baseline method using the LDA model using MALLET toolkit [16].

Ren *et al.* [17] focused on predicting users' opinions toward specific topics they had not directly given yet. They proposed the method based on a social context and topical context incorporated matrix factorisation framework.

2.2 Topic tracking

There are some studies that do not detect topics but track the topics down in chronological order. This section introduces the studies about topic tracking methods with latent topics.

Serizawa *et al.* [18] use LDA to extract topics and then judge and integrate the similarities between the topics to decide the appropriate number of topics. Based on the topic number, they propose a method to extract/track-down the topics.

In this study, we appropriately determine the topic change that can be found from the change of similarities of each month and analyse the features of the topics, including slang from the tendency of a topic change.

Zhao *et al.* [19] proposed a dynamic query expansion (DQE) model for theme tracking in Twitter. Their proposed model can express the theme consistency among heterogeneous entities through semantic and social relationships. They conducted the experiment for tracking the theme 'civil unrest.' As a result of that, they could confirm the effectiveness and scalability of DQE.

2.3 Neural topic modelling

Wan *et al.* [20] proposed a model that combined a deep neural network with a latent topic model.

Cao *et al.* [21] proposed a neural topic model and its supervised extension. They evaluated the performance of their neural topic models on the document-oriented supervised tasks including multi-class classification, multi-label classification and regression. As a result, their proposed method achieved higher accuracy than the other supervised topic modelling methods.

Larochelle *et al.* [22] proposed a model to learn semantic representations of texts from an unclassified text corpus. The evaluation experiments proved that their proposed model gives a good performance as a generative model of text documents and as a document representation learning algorithm.

2.4 Slang analysis

This section introduces several studies focusing on the analysis of unknown expressions, such as slang.

Hisano *et al.* [23] judge the similarity of proper nouns by using time-series correlation and word co-occurrence rate and then construct a similar word dictionary. With this dictionary, slangs that are quasi-words of proper nouns and abbreviations of proper nouns can become manageable.

The study by Hisano *et al.* extracts similar words by targeting a specific topic. However, we would like to deal with various kinds of slangs that are not only words expressing specific things, but also slang expressing general events such as 'posharu' (fizzle out) and words having both meanings as slang and as general words, such as 'chiiter' (cheater) and 'teppan' (a sure thing). Our study also differs by focusing on semantic elements such as usage of slang.

Matsumoto *et al.* [24] proposed the topic analysis based on latent topic modelling and word distributed representation. Their method analysed the change of topic or meaning of slang by using the topic keyword distributed vector and word distributed representation vector, which are calculated from the large tweet text corpus. As a result of the evaluation experiment, it was found that the change of slang's topic or meaning can be observed by using sentiment analysis and similarity calculation of topic modelling or word distributed representation. Their method focused on the topic change of positive and negative. However, the sentiment or impression of slang is often changed in a short period. So, we focus only on fluctuation of topic information of slang in a short period.

2.5 Sentiment and semantic analysis on big data

Ren *et al.* [25–27] proposed the sentiment and semantic analysis method on big data on the Internet. On the other hand, Matsumoto *et al.* [28] proposed the personality predicting method from users' tweets based on neural networks and sentence embedding. Sentiment analysis of web text or personality analysis of short text plays a very important role in many text mining tasks. Topic modelling should also be an effective and important component of sentiment analysis or personality analysis.

Ren *et al.* [29] proposed the emotion analysis method on social big data. Their method can obtain time-series emotion fluctuation for each Twitter users' cluster. They used word embedding method to extract text feature from short text on Twitter. A lot of short text data include noise, such as meaningless expressions or slangs. Many big data include such expression, so, we think that topic modelling is effective to extract meaning or sentiment indirectly.

3 Proposed method

This section describes the proposed method.

3.1 Equations

This section describes the procedure of similarity calculation between topics in each subsection, which is used in this study to investigate the change of topics. By tracking the fluctuation of topics, we analyse what meaning was used for each slang term.

Fig. 1 shows an example of calculating topic similarity and output of higher ranked topics regarding the word 'sweets' in April and May. The process of topic analysis is explained in order below.

3.1.1 Extraction of word set: First, the target months and slang are decided in the database, and then the wordlist is obtained from the target tuple. Next, the word set is made by extracting only nouns, unknown words, and slang from the wordlist.

In this process, the tweets that produced the completely matched word sets are regarded as the mechanically posted tweets and removed, leaving only the first-appeared tweets as a target for analysis.

3.1.2 Topic generation: Topics are generated by inputting the data of the reshaped word set. At this time, the hyper parameters are set as $\alpha = 0.5$, and $\beta = 0.1$, the number of words for each topic is set as 10, and the number of topics is set as 10.

Here, we explain how we decide the number of topics. As for the 40 kinds of the targeted Japanese slangs, we, respectively, prepare a corpus consisting of the tweets including each Japanese slang.

We calculate the LDA model perplexity (1) for each number of topics from 3, 4, 5, 6, 7, 8, 9 to 10.

The LDA model perplexity indicates the performance of model prediction and is defined by the inverse of probability. We use the number of topics that obtained the lowest averaged value of perplexity.

In this study, we set the number of topics as 10. To decide the parameter values of α and β, we calculated the average number of topics for each tweet and selected the combination of the parameters when the least number of topics was obtained. As this result, we set $\alpha = 0.5$ and $\beta = 0.1$

$$\text{perplexity}(D_{\text{test}}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log p(w_d)}{\sum_{d=1}^{M} N_d}\right\}, \tag{1}$$

where M is the number of documents, d is the document, w_d, is the words appeared in document d, N_d is the number of words in document d, and $p(w_d)$ is the appearance probability of word w_d.

The example result of the output is shown in Table 1. The result shows the word set of the slang 'sweets' in topic 2 in April.

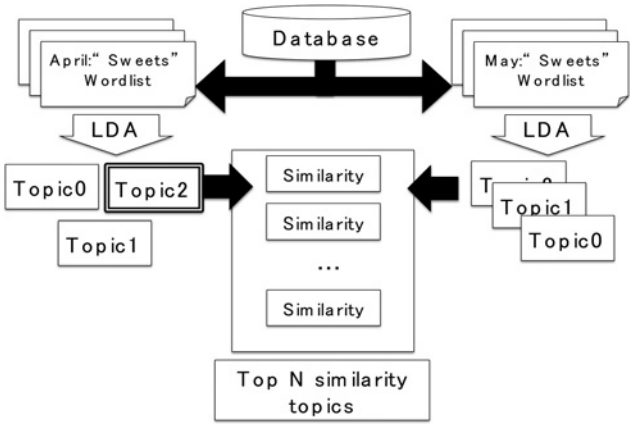

Fig. 1 *Calculation of topic similarity between April and May*

Table 1 Slang 'sweets' of topic 2 in April

Feature word	Appearance probability
laugh	0.0778
sweets	0.0742
girl	0.0438
series	0.0379
boy	0.0297
I	0.0156
thing	0.0108
woman	0.0103
previous	0.0101
it	0.0093

3.1.3 Similarity calculation: The topics are chosen to be used for analysis basis. Then, the similarities with the topics of other months are calculated, and the top 3 similarities are output. The values are calculated for each month so that the transitions of the top 3 similarities are analysed monthly.

When we calculate similarities, the vector consisting of the feature words and their appearance probabilities are used. However, because the tweets including the target slang are targeted for analysis, there is a problem in that the appearance probability of slang itself becomes very high. Generally, tf-idf is often used as a solution to this problem. Tf-idf is a value that multiplies term frequency by inverse document frequency.

In the topic model, the tf-idf value can be implemented by using $\text{term\_score}_{k,v}$ proposed by Blei *et al.* [30].

The calculation formula of $\text{term\_score}_{k,v}$ is shown by

$$\text{term\_score}_{k,v} = \hat{\beta}_{k,v} \log\left(\frac{\hat{\beta}_{k,v}}{\left(\prod_{j=1}^{K} \hat{\beta}_{j,v}\right)^{1/K}}\right), \tag{2}$$

where $\hat{\beta}_{k,v}$ is the appearance probability of v in topic k and K is the number of topics ($k \in \{1,..., K\}$).We weight the term appearance probability using $\text{term\_score}_{k,v}$. After that process, we decide the $\text{term\_score}_{k,v}$ corresponding to the feature words as the vector $\boldsymbol{x}$ in that topic and calculate cosine similarities between the topics with (3). The calculated cosine similarities are used for analysis as the similarities between the topics

$$\cos\left(\boldsymbol{x}_i \cdot \boldsymbol{x}_j\right) = \frac{\boldsymbol{x}_i \cdot \boldsymbol{x}_j}{\left|\boldsymbol{x}_i\right|\left|\boldsymbol{x}_j\right|}. \tag{3}$$

3.1.4 Latent Dirichlet allocation: To analyse the usage of slang, we focus on the analysis of the latent topics. In this study, we analyse the latent topics by using LDA.

LDA is a probabilistic topic model that assumes multiple topics exist in a document. The graphical model of the LDA and the explanation of each variable are shown in Fig. 2 and Table 2.

The steps of document generation by LDA is as follows. Dir indicates Dirichlet distribution, and Multinomial indicates multinomial distribution.

In this study, we calculate LDA by using the tool: GibbsLDA++ [31].

(i) The distribution of topic θ_i–Dir(α) is generated for each document $i \in \{1,..., M\}$.
(ii) The distribution of word ϕ_k –Dir(β) is generated for each topic $k \in \{1,..., K\}$.
(iii) The topics z_{dn} –multinomial (θ_i) are generated for $i \in \{1,..., M\}$ and each word $j \in \{1,..., N_i\}$.
(iv) Words w_{dn}–multinomial ($\varphi_{z_{i,j}}$) are generated.

3.1.5 Database: This study targets the tweet sentences including slang for analysis. Since we would like to focus on the time-series analysis, tweet sentences need to be collected for each month.

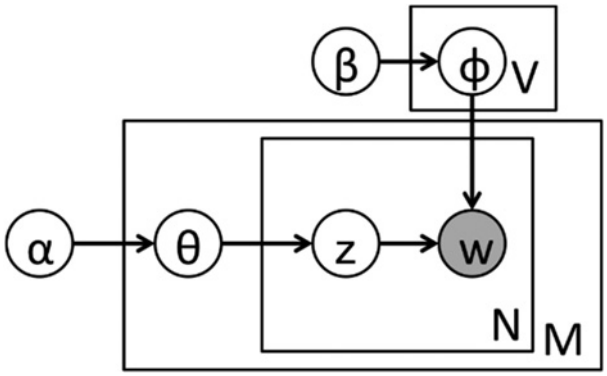

Fig. 2 *LDA graphical model*

Table 2 Parameters of LDA

α	topic occurrence parameter
β	word occurrence parameter of topic K
K	number of topics
M	number of documents
N	number of words in document M
θ	topic distribution in document M
Z	topic of word n in document M
W	word

Therefore, we construct a database to more effectively use the collected documents. Table 3 shows the structure of each table. It consists of four columns: tweet posting date (year, month, day, hour, minute, second), the account ID of the user who posted the tweet, tweet sentences, and the data with format of <Word/POS> obtained by morphologically analysing tweet sentences with MeCab [32].

At this time, the 654 slang words are already registered in a user dictionary in MeCab and the '-unk-feature UNK' is set as an option. Currently, we use SQLite3 [33] as the database-management system, and 148,551,674 tweets are registered into the database. We would like to update the database in the future.

3.2 Topic fluctuation analysis

This subsection describes the method to calculate scores for each topic by month. In this process, we use the same database, wordlist, and LDA as those that are used in the previous section. Fig. 3 shows the system flow.

3.2.1 Extraction of word set: We select the slang from the database and obtain the wordlist from the target tuple. Next, we make the word set by extracting only nouns, unknown words, and slang from the wordlist.

During this process, we remove the tweets that produced the word sets completely matched to the other tweets' word sets from the analysis target. As the stop words, we remove the words that do not include kanji, hiragana, or katakana characters because these words are not Japanese.

3.2.2 Topic generation: We generate topics by using all wordlists of each month as input data. During this process, we set the hyper parameters as $\alpha = 0.5$, $\beta = 0.1$, and the number of words for each topic is set as 100 and the number of topics is set as 100.

3.2.3 Topic clustering: If the number of topics is set as 100, a lot of similar topics will be generated. Therefore, after the analysis by

Table 3 Structure of the table in the slang tweet database

Field name	time	nameID	Tweet	wordlist
data type	date time	text	text	text

CAAI Trans. Intell. Technol., 2019, Vol. 4, Iss. 1, pp. 64–71

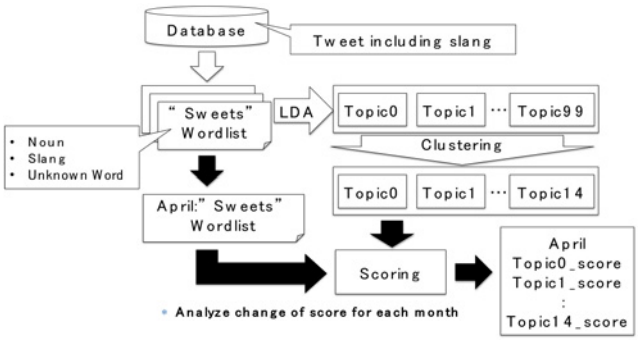

Fig. 3 *System flow (topic fluctuation analysis)*

LDA, we reduce the number of topics by using the unsupervised clustering method.

After the clustering, we use the centroid vector of each cluster as the representative vector of each topic. In this step, the number of clusters is automatically decided by setting the threshold of similarities between vectors.

3.2.4 Topic scoring: We obtain the wordlist for each month and annotate the score for each topic. The equation of topic scoring is shown by

$$\text{topic-score}_{m,k} = \frac{\sum_{i=1}^{\text{word}_k} \left(\hat{\beta}_{k,i,v} \times \text{WordCount}_{m,i}\right)}{\text{TweetCount}_m}, \quad (4)$$

where $\hat{\beta}_{k,i,v}$ is the probability of occurrence v of word i in topic k; TweetCount_m is the number of tweets in month m; $\text{WordCount}_{m,i}$ is the number of tweets in which word i appeared in month m; and word_k is the feature word in the topic k.

At the time of scoring the tweets, if the feature words appeared in the tweet sentences, we add the corresponding appearance probability as a feature quantity. Finally, the averaged value calculated by the number of tweets is defined as the score of the topic.

Since the score can be represented by multiplying the appearance probability of the target word and the number of tweets including the target words, the calculation is the same as (3). In Experiment 2, we annotate the score for each topic by using this scoring method and output the monthly fluctuation.

3.3 Comparison with tweet distributed representation

One of the problems regarding the analysis on tweet topic change by the topic vector is that the relation of word co-occurrence cannot be obtained sufficiently because a tweet (document), which is targeted for analysis, is short. Besides, in simple bag-of-words model, weights such as tf-idf are calculated on words to measure word importance value. However, this causes too high importance values on rare expressions appearing in only specific topics.

On the other hand, there is a study reporting that the word distributed representation and the document representation learning are more effective to extract short text feature. Liu *et al.* [34] proposed the task-oriented word embedding for the text classification method. From the results of evaluation experiments, it was found that their approach significantly outperformed the state-of-the-art methods.

Boom *et al.* [35] found that high performance is shown in the short text representation learning task by weighting word distributed representation with idf value.

As in their study, it is effective to learn text representation based on word distributed representation. In this section, we create a model that generates a fixed dimensional real-valued vector by training distributed representations of tweets. We extract a 64-dimensional vector from tweets based on this model.

We generate a cluster vector for each month with consideration of cluster appearance frequency and similarity, by conducting clustering on the extracted vector set.

We assume that this vector should express the meaning on the tweet set. We calculate the similarity between the cluster vectors obtained from the tweet set of the neighbouring month and detect the semantic change of the slang by finding out when the similarity becomes lower than the average similarity. We also use topic vector similarity to analyse the tendency appeared.

Since the number of tweets is different in each month, we limit the number of tweets up to 100 and create a distributed representation cluster vector by randomly extracting the 100 tweets from each month. To learn the tweet distributed representations, we use pre-trained word distributed representation as a feature in this study.

We use the nouns extracted from each tweet as the training data, and trained 64-dimensional distributed representations by word2vec [36] using the skipgram algorithm and setting context window size as 5.

We randomly extracted one million tweets as target tweets. Then, the words in each tweet are replaced by the 64-dimensional distributed representation vector. For each tweet, vector up to 32 words are input, and the encoder–decoder based on convolutional neural networks is learned. From the hidden layer of the learned encoder–decoder, the 64-dimensional vector responding to the inputted tweet. This network model is hereinafter called as CNN-AE. Fig. 4 shows the layer construction of CNN-AE.

We used ReLU as an activation function between the hidden layers and Softmax as an activation function for the output layer. As the optimisation algorithm, we used Adam and set the mini batch size as 256 and learning epochs as 1000.

The output of the fifth layer (hidden layer) is extracted as the 64-dimensional vector. We conduct clustering on the tweets with the k-means method by using the trained model. The number of clusters was set as 60, 100, 200, 300, 400 and 500.

We assume that the target slang should have its own tendency to appear in each independent topic. Therefore, if several different slangs co-occurred in the same tweet in the same cluster, we can regard that classification by topic should have been failed. We judge the result of the clustering result according to the average value of the co-occurrence frequency between the different slangs in the same cluster. When the value is smaller, we judge it as better clustering result. As a result of the clustering, the optimal number of clusters was $N = 200$.

Therefore, in this study, we set $N = 200$ and analyse the semantic change of slangs by using the feature vector (semantic feature

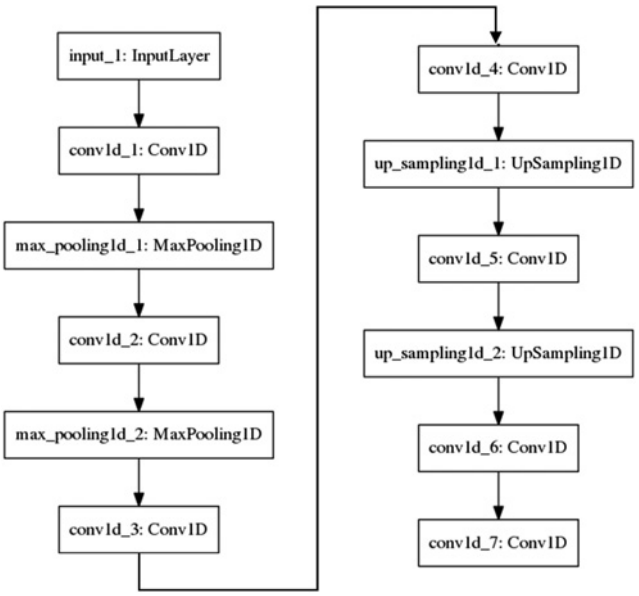

Fig. 4 *CNN-AE: auto encoder–decoder*

Table 4 Part of the target Japanese slangs

boomerang	oni (devil)	gobaku (bombing error)	enban (disk)	yuri (lily)
Jizou (a stone statue of Jizo)	kami (god)	teppan (grill)	atsui (hot)	pizza
urayama (a mountain at the back)	daishouri (big winning)	Gunnmar (people of Gunnma prefecture)	daisougen (prairie)	anntei (stabilisation)
sweets	kuudere	jiwaru	Youtsube (YouTube)	denntotsu

vector), which is created based on the appearance frequency in each month of the tweets belonging to each cluster instead of a topic vector.

4 Experiments

This section describes the experiment of the proposed method.

4.1 Analysis of topic change: experiment 1

We investigated the transition of each topic score in the term from April 2015 to December 2015 by using the method described in the previous section. We randomly selected a basis topic from the topics in April and calculated similarities with the topics from May 2015 to September 2015.

In this study, we target 40 kinds of Japanese slangs shown in Table 4.

The topic similarity was calculated by using cosine similarity between the vectors of topics, of which dimension is word and value is occurrence probability.

As a result, we found the tendency of some words. The examples are described in the following subsection.

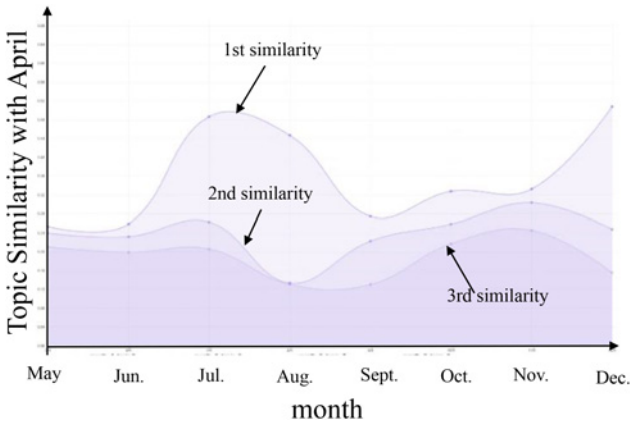

Fig. 5 *Similarity fluctuation in topic 0 of slang* '*boomerang*'

4.1.1 Topic occurrence:'boomerang':

Fig. 5 shows the score transition which was obtained by analysing the slang word 'boomerang.' A graph is displayed with an analysis result visualisation tool which was created by a graph plot library of JavaScript called Chart.js [37].

From this figure, we can see that the similarities are increasing from June to July. Therefore, we focused on the topics that calculated the highest similarity from June to September. Table 5 shows the feature words of each topic.

From this table, 'boomerang' in topic 0 in April means that the calumny or criticism towards other people goes for the persons themselves who posted such words.

This indicates that the word 'boomerang' is used as slang in a so-called 'boomerang statement.' In fact, the word 'boomerang' co-occurs with the word 'statement' in the topic.

Next, we focus on the topics from June and July. In June, 'boomerang' was not used as slang in meaning. However, in July, 'boomerang' was found to be used as slang.

On the other hand, 'boomerang' as slang appeared in the topic of August. However, in the topic of September when the similarity decreased, the latent topic of 'boomerang' cannot be judged by a human. As a result, it was found that the topic fluctuation occurred by topic similarity change.

Moreover, the topic keywords are thought to occur among the topic basis when the similarity increases. The topic keywords are thought to fall into decline when the similarity decreases.

4.1.2 Topic continuance: 'sweets':

Next, Fig. 6 shows the transition of similarity when the topic 2 of the slang word 'sweets' was set as a basis.

From this figure, it is found that the similarities repeatedly increased and decreased. By focusing on the topics which obtained the highest similarity in the period from June to September, the feature words for each topic are shown in Table 6.

The word 'sweets' as slang has two meanings: (i) the sweet dessert and (ii) an ironic meaning regarding people who regurgitate information by mass medium or bandwagon in the world.

From this table, 'laugh', 'sweets', and 'girl' are ranked higher in every month so that the latent topic can be confirmed as the slang 'sweets' having ironic meaning.

4.2 Analysis of topic score transition: experiment 2

4.2.1 Topic score transition: 'boomerang':

The score transition obtained by analysing slang 'boomerang' is shown in Fig. 7. We found that the highest score was calculated in May.

It is also found that the score of topic 3 increased until December. So, we investigated what words are the feature words in topic 8 and topic 3. A part of the feature words in topic 3 is shown in Table 7.

In the set of tweets in December, the tweet with the highest score of topic 3 is defined as Example sentence 3.

> Example sentence 3: Tweet sentence belonging to topic 3
> 'Because they say that specialised cat for ***boomerang*** is strong, I try to select it carefully.'

Table 5 Similar topics of slang 'boomerang' April, topic 0 as a basis

Apr. (T0)	Jun. (T3)	Jul. (T2)	Aug. (T7)	Sep. (T6)
boomerang	*boomerang*	*boomerang*	thing	*boomerang*
myself	laugh	thing	people	thing
I	partner	laugh	*boomerang*	myself
yo	weapon	myself	myself	*yo*
what	head	I	*minshuto*	people
video	myself	flaming	*yo*	child
statement	up	statement	what	I
all	what	*kuse*	statement	face
flaming	change	*mon*	congressmen	outside
san	I	number	criticism	story

CAAI Trans. Intell. Technol., 2019, Vol. 4, Iss. 1, pp. 64–71

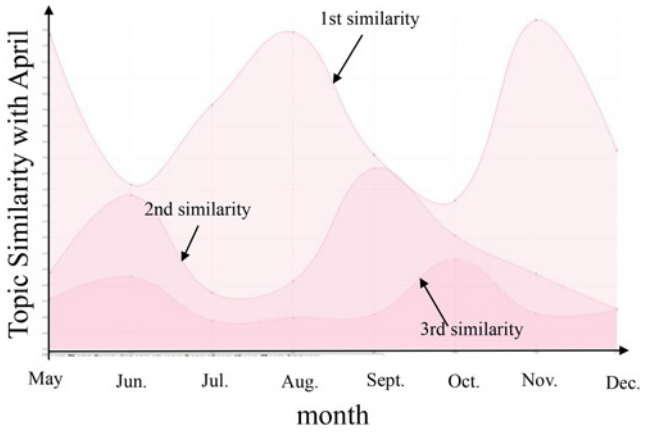

Fig. 6 *Similarity transition of slang 'sweets' in topic 2*

Table 6 Similar topics with topic 2 of slang 'sweets' in April

Apr. (T2)	Jun. (T9)	Jul. (T5)	Aug. (T1)	Sep. (T6)
laugh	laugh	laugh	laugh	sweets
sweets	sweets	sweets	sweets	laugh
girl	girl	girl	girl	girl
series	like	series	series	people
boy	thing	'symbol'	boy	thing
I	people	boy	people	like
thing	focus	another stomach	*kun*	focus
woman	woman	power	story	power
previous	myself	stomach	man	I
it	*yo*	yeah	like	sweets (laugh)

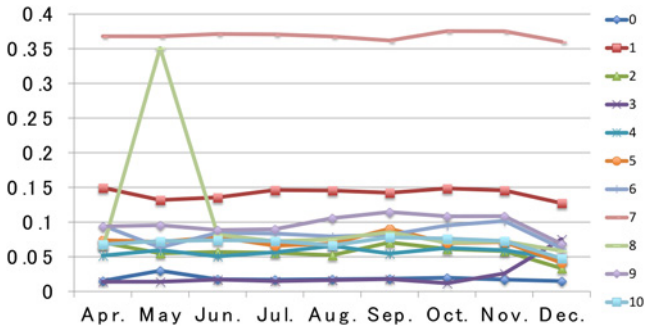

Fig. 7 *Score transition of slang 'boomerang'*

Table 7 Feature words in topic 3 of slang 'boomerang'

Feature word	Score
weapon	0.373684
nyanta-ka	0.357511
penetration	0.297841
attack	0.273524
cat	0.25364
sword	0.249143
	0.188299

4.2.2 Topic score transition: 'reiyar': The transition of the scores when the slang 'reiyar' was analysed is shown in Fig. 8. 'Reiyar' means 'people who do costume play.'

From Table 6 and the example tweet sentences, it was found that the topics of games are generated. In these instances, 'boomerang' was used with the literal meaning instead of the slang meaning. Therefore, we could not confirm the difference in how the topics occur when the word was used with both slang meaning and literal meaning.

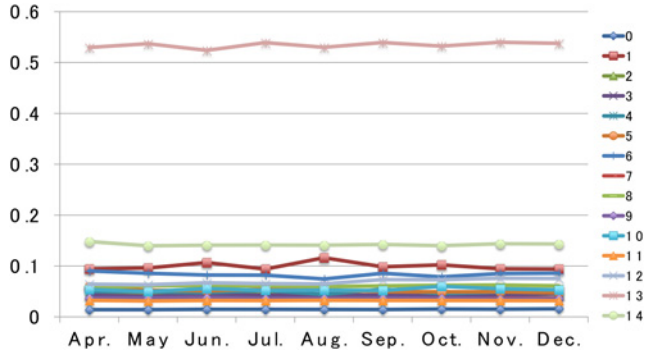

Fig. 8 *Score transition of slang 'reiyar'*

In future analysis, we need to compare slang words that do not have any meaning other than the slang meaning, such as 'yukadon' etc. We also need to compare the change of meanings of the words that do not have slang meanings. As seen in Fig. 7, the transition of the scores in each month could almost not be confirmed. This suggests that the usage of 'reiyar' did not change or the slang was not used beyond one year. Therefore, it will be necessary to investigate changes over a longer period of time.

4.2.3 Tweet embedding approach: experiment 3: In this subsection, we compare the change detection method by topic similarity with the method by tweet embedding cluster frequency.

To detect the changes, we look at whether the similarity between the feature vectors in each neighbouring month is over the average value or not, and used it as the basis.

The followings are the parameters used in this experiment:

- The number of topics for LDA: 60.
- The number of the clusters for tweet embedding: 200.

Figs. 9–12 show the result of the comparison between the result by LDA ('topic') and the result by tweet embedding ('embedding') as for the change of the similarities.

The left vertical axis indicates the similarity based on 'topic' and the right vertical axis indicates the similarity based on 'embedding.' The horizontal axis indicates the interval of the similarity calculation. For example, '4_5' indicates the similarity between feature vectors in April and May.

As we can see from these results, there are some slangs whose similarities did not change so much, however, most slangs showed changes of similarities differently depending on the methods.

In each method, the averaged similarity values of the all intervals is, respectively, expressed as $\overline{sim}_{topic}$ and $\overline{sim}_{emb}$, and the similarity values in each interval (i) is expressed as sim_{topici}, sim_{emb_i}. We define $match_i$ as (5) that takes 1 when the value of similarity is

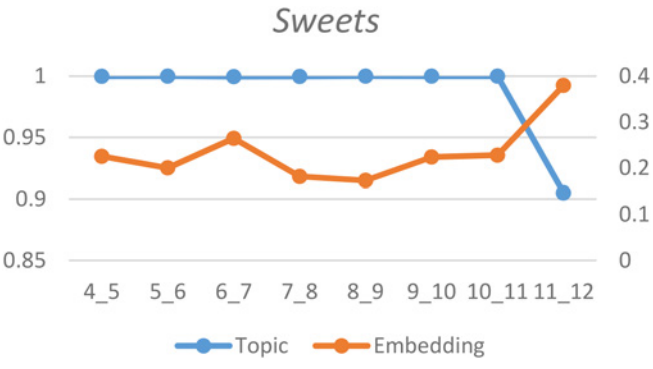

Fig. 9 *Similarity transition of 'sweets' between neighbour month for each feature extraction method*

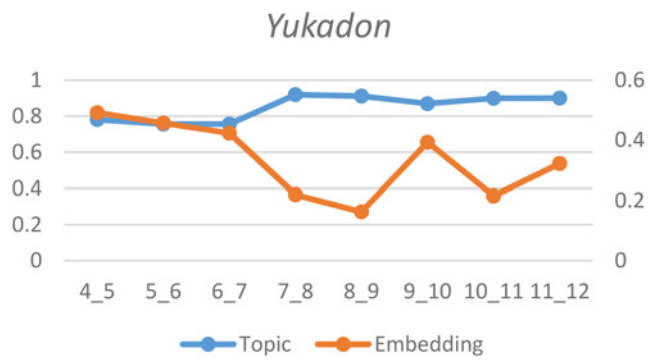

Fig. 10 *Similarity transition of 'yukadon' between neighbour month for each feature extraction method*

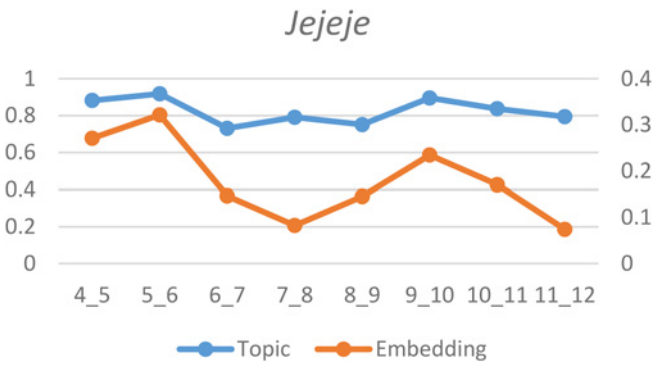

Fig. 11 *Similarity transition of 'jejeje' between neighbour month for each feature extraction method*

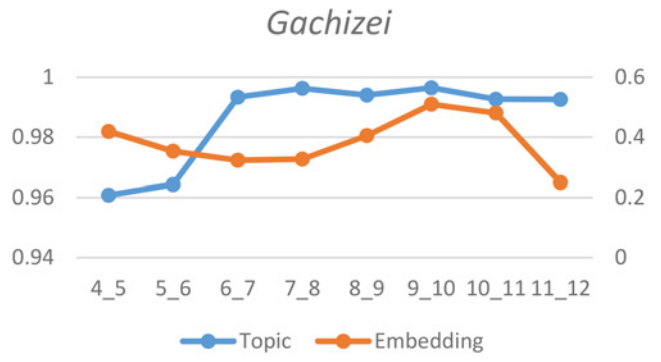

Fig. 12 *Similarity transition of 'gachizei' between neighbour month for each feature extraction method*

under the average in both the methods and that takes 0 when the value of similarity is over the average in both the methods

$$\text{match}_i = \begin{cases} 1 & \text{if } \text{sim}_{\text{topic}_i} < \overline{\text{sim}_{\text{topic}}} \text{ and } \text{sim}_{\text{emb}_i} < \overline{\text{sim}_{\text{emb}}}, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The average value of match_i is defined as (6). M stands for the number of intervals

$$\overline{\text{match}} = \frac{\sum_{i=1}^{M} \text{match}_i}{M}. \quad (6)$$

When we calculated $\overline{\text{match}}$ by (6) for all the target slangs and obtained the average, the value was 0.446, which was under 50%. This result shows that the tweet embedding extracts feature by different viewpoint from the topic model.

The slang 'Jejeje' is a buzz term used on a TV drama, and it is usually used to express a surprise. This expression has wide application range, however, in our tweet corpus, this slang seems to be used often in the topics related to that specific drama, therefore, the results of both methods (topic and tweet embedding) are similar.

5 Discussions

To extract features of slang, which is the purpose of our study, we focused on the relationship between topic change and similarity transition in experiment 1. It is therefore important to determine whether we can consider this relationship as a slang feature.

For example, we think that we can clarify what features slang has by investigating whether declination of topics is limited to the topics focusing on slang. However, there were some problems with the method we used in Experiment 1.

First, the topics serving as a basis were manually selected. Therefore, a method to automatically select topics is required.

Second, change of topics was judged manually as was the tendency obtained by Experiment 1. Different tendency could be determined depending on how the judging person observes it. It would not be practical to classify them mechanically.

In Experiment 2, we generated topics, not from the change of each month, but from the whole data set and analysed the usage of each topic by month. We could observe the case where the score transition almost did not occur and the case where score change considerably occurred.

For example, if the topics are sorted out in order of high score and the values indicating the change of rank for each month are obtained, we would be able to classify slang words into one group of slang that is rarely changed and another group of slang that is frequently changed. By fixing the conditions of change to quantify the feature of slang, we think that Experiment 2 could be used for slang classification.

In Experiment 3, we found several differences between the topic modelling and the tweet embedding. Each method can capture different feature of slang. For example, the topic modelling can express the real-time meaning of the slang, and the tweet embedding can express the general meaning of the slang. If we can use these two types of features at the same time, we think that could obtain more detailed fluctuations of the slang meanings from social media.

6 Conclusion

In this study, we conducted two experiments to consider slang features. In Experiment 1, we investigated topic change by generating a topic model for each month by LDA and analysing transition of each similarity. As a result, we could observe the decrease/increase in the occurrence of topics when the similarity increased/decreased. However, because there are a lot of human interventions in these processes, we need to propose an automatic derivation method.

In Experiment 2, we generated the topic model from the whole data set and calculated each topic score for each month. We could extract topic occurrence by observing the transition of the scores.

With this method, it is expected that we can classify slang based on the change of topics or the number of topics. Therefore, we think that that this provides a clue for an automatic slang detection method if we develop the study based on Experiment 2.

In Experiment 3, we compare two types of slang feature extraction: topic modelling and tweet embedding. As the evaluation result, we found that the topic modelling and tweet embedding can extract different meanings or topics.

As a future task, we would like to cluster slang from the change of scores obtained in Experiment 2. After that, we would need to investigate what words are found in each cluster.

CAAI Trans. Intell. Technol., 2019, Vol. 4, Iss. 1, pp. 64–71

At the present stage, 40 slang terms have been output as results of our analysis. The transition of scores and the feature word in each topic are described in the Appendix. We would like to propose a slang clustering method based on these data.

Additionally, we would like to add tweet sentences into the database at any time, attempt to calculate similarity in a longer period, and focus on finer changes by changing the time range from month unit into day unit.

7 Acknowledgment

This work was supported by JSPS KAKENHI under grant numbers JP15K16077 and JP15H01712.

8 References

[1] Blei, M.D., Ng, Y.A., Jordan, M.I.: 'Latent Dirichlet allocation', *J. Mach. Learn. Res.*, 2003, **3**, pp. 993–1022
[2] Bolelli, L., Ertekin, Ş., Giles, C.L.: 'Topic and trend detection in text collections using latent Dirichlet allocation'. European Conf. on Information Retrieval ECIR 2009: Advances in Information Retrieval, Grenoble, France, 2009, pp. 776–780
[3] Huang, J., Peng, M., Wang, H., *et al.*: 'A probabilistic method for emerging topic tracking in microblog stream', *World Wide Web*, 2017, **20**, (2), pp. 325–350
[4] Lau, J.-H., Collier, N., Baldwin, T.: 'On-line trend analysis with topic models: #twitter trends detection topic model online'. 24th Int. Conf. on Computational Linguistics COLING 2012, Mumbai, India, 2012, pp. 1519–1534
[5] Hong, L., Davison, B.D.: 'Empirical study of topic modeling in twitter'. First Workshop on Social Media Analytics, Washington, DC, USA, 2010, pp. 80–88
[6] Wang, Z., Iwaihara, M.: 'Cross-lingual tweet recommendation based on user interest using bilingual LDA'. DEIM Forum 2015, Fukushima, Japan, 2015, A8-2
[7] Grant, C., George, C.P., Jenneisch, C., *et al.*: 'Online topic modeling for real-time twitter search'. 20th Text REtrieval Conf. (TREC2011), Maryland, USA, 2011
[8] Qiang, J., Chen, P., Wang, T., *et al.*: 'Topic modeling over short texts by incorporating word embeddings', *arXive preprint arXiv:1609.08496v1*, 2016
[9] Valeriano, E., Juanjosé, L., Peña, T., *et al.*: 'An LDA-lexical syntactical approach for events and features extraction of earthquakes from Spanish and English tweets'. Fourth Annual Int. Symp. on Information Management and Big Data, SIMBig 2017, Lima, Peru, 2017
[10] Grün, B., Hornik, K.: 'Topicmodels: an R package for fitting topic models', *J. Stat. Softw.*, 2011, **40**, (13), pp. 1–30
[11] Sievert, C., Shirley, K.E.: 'LDAvis: a method for visualizing and interpreting topics'. Workshop on Interactive Language Learning, Visualization, and Interfaces, Baltimore, Maryland, USA, 2014, pp. 63–70
[12] Amoualian, H., Gaussier, E., Clausel, M., *et al.*: 'Streaming-LDA: a copula-based approach to modeling topic dependencies in document streams'. 22nd ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, San Francisco, United States, 2016, pp. 695–704
[13] Alghamdi, R., Alfalqi, K.: 'A survey of topic modeling in text mining', *Int. J. Adv. Comput. Sci. Appl.*, 2015, **6**, (1), pp. 147–153
[14] Kimura, T., Miyamori, H.: 'A method of classifying relationships between hashtags using co-occurrence and latent topics', *J. IEICE*, 2015, **J98-D**, (8), pp. 1151–1161, doi: 10.14923/transinfj.2014JDP7142
[15] Hashimoto, K., Kontonatsios, G., Miwa, M., *et al.*: 'Topic detection using paragraph vectors to support active learning in systematic reviews', *J. Biomed. Inf.*, 2016, **62**, pp. 59–65
[16] MALLET toolkit. Available at http://mallet.cs.umass.edu/
[17] Ren, F., Wu, Y.: 'Predicting user-topic opinions in twitter with social and topical context', *IEEE Trans. Affective Comput.*, 2013, **4**, (4), pp. 412–424
[18] Serizawa, M., Kobayashi, I.: 'Topic tracking attempt by considering the number of topic in document'. 2012 Annual Meeting of the Association of Natural Language Processing, Hiroshima, Japan, 2012, pp. 1196–1199
[19] Zhao, L., Chen, F., Lu, C.-T., *et al.*: 'Dynamic theme tracking in twitter'. 2015 IEEE Int. Conf. on Big Data (Big Data), Santa Clara, CA, USA, 2015
[20] Wan, L., Zhu, L., Fergus, R.: 'A hybrid neural network-latent topic model'. 15th Int. Conf. on Artificial Intelligence and Statistics (AISTATS), La Palma, Canary Islands, 2012, pp. 1287–1294
[21] Cao, Z., Li, S., Liu, Y., *et al.*: 'A novel neural topic model and its supervised extension'. Twenty-Ninth AAAI Conf. on Artificial Intelligence (AAAI-15), Texas, USA, 2015, pp. 2210–2216
[22] Larochelle, H., Lauly, S.: 'A neural autoregressive topic model', *Adv. Neural. Inf. Process. Syst.*, 2012, **4**, pp. 2708–2716
[23] Hisano, Y., Sawase, K., Nobuhara, H.: 'Extraction of similar words based on adaptation and time-correlation of maximal substrings from tweets of the same topic'. IEICE Technical Report, SIS2012-49, 2013
[24] Matsumoto, K., Yoshida, M., Tsuchiya, S., *et al.*: 'Slang analysis based on variant information extraction focusing on the time series topics', *Int. J. Adv. Intell.*, 2016, **8**, (1), pp. 84–98
[25] Ren, F., Kang, X., Quan, C.: 'Examining accumulated emotional traits in suicide blogs with an emotion topic model', *IEEE. J. Biomed. Health. Inform.*, 2016, **20**, (5), pp. 1384–1396
[26] Ren, F., Yu, H.: 'Role-explicit query extraction and utilization for quantifying user intents', *Inf. Sci.*, 2015, **329**, (1), pp. 568–580
[27] Ren, F., Matsumoto, K.: 'Semi-automatic creation of youth slang corpus and its application to affective computing', *IEEE Trans. Affective Comput.*, 2015, **7**, (2), pp. 176–189
[28] Matsumoto, K., Tanaka, S., Yoshida, M., *et al.*: 'Ego-state estimation from short texts based on sentence distributed representation', *Int. J. Adv. Intell.*, 2017, **9**, (2), pp. 145–161
[29] Ren, F., Matsumoto, K.: 'Emotion analysis on social big data', *ZTE Commun.*, 2017, **15**, (S2), pp. 30–37
[30] Blei, D.M., Lafferty, D.J.: 'Text mining: theory and applications, chapter topic models' (Taylor and Francis, UK, 2009)
[31] GibbsLDA++. Available at https://github.com/mrquincle/gibbs-lda
[32] MeCab. Available at http://taku910.github.io/mecab/
[33] SQLite3. Available at https://www.sqlite.org/index.html
[34] Liu, Q., Huang, H., Gao, Y., *et al.*: 'Task-oriented word embedding for text classification'. 27th Int. Conf. on Computational Linguistics, New Mexico, USA, 2018, pp. 2023–2032
[35] Boom, D.C., Canneyt, V.S., Demeester, T., *et al.*: 'Representation learning for very short texts using weighted word embedding aggregation', *Pattern Recognit. Lett.*, 2016, **80**, pp. 150–156
[36] Mikolov, T., Sutskever, I, Chen, K., *et al.*: 'Distributed representations of words and phrases and their compositionality'. 26th Int. Conf. on Neural Information Processing Systems (NIPS'13), Lake Tahoe, Nevada, USA, 2013
[37] Chart.js. Available at https://www.chartjs.org/