International Journal of Advanced Intelligence
Volume 9, Number 2, pp.145-161, May, 2017.
© AIA International Advanced Information Institute

AIA
International
Advanced
Information
Institute

Ego-state Estimation from Short Texts Based on Sentence Distributed Representation

Kazuyuki Matsumoto, Satoshi Tanaka, Minoru Yoshida, Kenji Kita and Fuji Ren
Graduate School of Technology, Industrial and Social Sciences, Tokushima University
Minamijosanjima cho 2-1, Tokushima, Japan

Received (10 Sep. 2016)
Revised (15 Apr. 2017)

Human personality multilaterally consists of complex elements. Egogram is a method to classify personalities into patterns according to combinations of five levels of ego-states. With the recent development of Social Networking Services (SNS), a number of studies have attempted to judge personality from statements appearing on various social networking sites. However, there are several problems associated with personality judgment based on the superficial information found in such statements. For example, one's personality is not always reflected in every statement that one makes, and statements are influenced by a personality that tends to change over time. It is also important to collect sufficient amounts of statement data including the results of personality judgments. In this paper, to produce an automatic egogram judgment, we focused on the short texts found on certain SNS sites, especially microblogs. We represented Twitter user comments with a distributed representation (sentence vector) in pre-training and then sought to create a model to estimate the ego-state levels of each Twitter user using a deep neural network. Experimental results showed that our proposed method estimated ego-states with higher accuracy than the baseline method based on bag of words. To investigate changes of personality over time, we analyzed how the match rates of the estimation results changed before/after the egogram judgment. Moreover, we confirmed that the personality pattern classification was improved by adding a feature expressing the degree of formality of the sentence.

Keywords: Egogram, Personality Estimation, Twitter, Social Networking Service, Distributed Representation, Deep Neural Network

1. Introduction

With the development and spread of Internet technology and the proliferation of information and communication terminals, social networking services (SNS) have become places to form or express our personalities, providing countless opportunities to offer our opinions. If a user's personality as expressed on SNS can be estimated, we might be able to use the result for information recommendations or marketing surveys. The techniques of personality estimation and personality representation are particularly essential if, in the future, we are to create communicating robots—for example, nursing robots—with the ability to respond and behave like humans.

Personality analysis from the linguistic information of individuals has a long history of study. Because personality judgments have historically required professional, complicated judgment, various questionnaire-based methods have been developed to facilitate the task of making these judgments. The Egogram test and the Big Five personality test are well-known examples. However, these tests tend to lack objectivity since the person who is given the judgment answers the questionnaire. We believe that one potential way to better achieve objectivity would be to judge personality based on an individual's statements on weblogs or SNS. In Section 2 of this paper, we introduce related works on judging personality based on SNS, then describe how these studies differ from ours. Section 3 describes the egogram that we targeted. Section 4 explains the proposed method, and Section 5 discusses the method and its results through an analysis of our evaluation experiment. Section 6 offers conclusions.

2. Related Works

Social Networking Services (SNS) such as Twitter and Facebook have become prime media for presenting information for and from individuals. This phenomenon has encouraged a number of studies which have attempted to estimate personality from the information opened on social media by a user [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]. An international workshop related to personality analysis has been held 4 and studies seeking to address personality from an engineering viewpoint have continued to increase in number and scope. Personality investigation is considered important for marketing surveys as well as vocational attitude testing.

IBM[12] and SynergyMarketing[13] have developed large scale questionnaires for such purposes. The results of these investigations have demonstrated that individual features, including personality, have strong relevance to purchasing behavior and preference. As a consequence, global-scale databases are being constructed.

Our study focuses on using the tweets of Japanese Twitter users living in Japan as a basis for analyzing and judging personality. Three previous studies are closely related to ours. Minamikawa et al. [8], [9] proposed a method to estimate egograms from Japanese weblogs using Multinomial Naive Bayes as a machine learning algorithm for estimation. Their proposed method refined features by calculating information gain, achieving an accuracy rate more than 20% higher than the baseline method using the bag-of-words feature.

Nasukawa et al.[11] analyzed linguistic features of Japanese-language authors as they relate to personality by adapting the textual analysis application LIWC (Linguistic Inquiry and Word Count). In their study, features corresponding to 79 word categories were extracted from the tweets of Japanese Twitter users using a dictionary constructed by converting LIWC into Japanese. The study analyzed the features by focusing on the correlation between personality profiles and word categories.

Okumura et al. [7] used an emotion judgment system to compare the pseudopersonalities estimated from impressions of weblog articles and results from the NEO-FFI test. The premise was that if the difference between pseudo-personality and real personality could be judged, the evaluator would be able to analyze mental states (e.g., detect depression).

Our study has marked differences from the studies cited here. As described above, Nasukawa et al. analyzed personality by using a Japanized LIWC. However, their method is unable to analyze words that are not registered in the LIWC. Consequently, buzz words or slang words that are often used in spoken language are excluded from the analysis. In our study, we use features obtained by converting all words included in a sentence into distributed expressions without reliance on the categorization of the LIWC.

While Okumura et al. produced an emotion judgment from tweets for their analysis, we do not. Although emotion appears to be deeply related to personality, there are individual differences in emotional expressions in tweets, and these individual differences can affect the accuracy of the emotion judgment. In our study, by directly extracting features from the sentence, we attempt to analyze personality without using such ambiguous and impressionable information as emotion that can be affected by individual differences.

In addition, our study uses the egogram as the basis for determining personality, while Okumura et al. used the Big Five test[14]. The Big Five test generally addresses deep-layer mental features and, as such, is suitable for analyzing personality dimensions that tend not to change in an individual ’ s lifetime. In contrast, by using the egogram, we believe that we are able to not only analyze personality but also support the possibility of individuals having a special chemistry with one another, and, additionally, to provide advice for actions and thoughts needed for self-reformation to change into the desired egogram (ideal personality).

3. Egogram

The Egogram[15] is a personality assessment tool proposed by J. M. Dusey that analyzes personality based on five ego-states: CP (Critical Parent), NP (Nurturing Parent), FC (Free Child), AC (Adapted Child), and A (Adult). By analyzing a graph illustrating these five ego-state levels, we are able to judge personality or compatibility. We used the Tokyo University Egogram Ver.2: TEG2[16] as the standard, choosing not to use the Big Five test.

In their analysis, Saito et al. [17] used the Big Five test, which, as its name implies, assesses features of personalities based on five main characteristics. These five characteristics are different from the ego-states used in an egogram and express the mental features of individuals. Given the difficulty of a third party to know these features and the difficult task of estimating these features from superficial features such as the contents of an utterance, we chose to use the egogram, as we were convinced that his would allow a third party to make almost the same judgment. Table 1 explains the five ego-states used in an egogram. In our study, we use the following letter scheme to indicate the level of each ego-state: ‘a’ denotes high, ‘b’ denotes neutral and ‘c’ denotes low. These letter designations are recorded in the order of CP, NP, A, FC, AC to indicate a particular personality pattern (for example, ‘abaac’). In all, there are $3^5 = 243$ possible patterns. Each pattern can be further divided into smaller patterns. Table 2 shows the egogram pattern classifications.

Table 1. Five ego-state in Egogram

Ego-state	High	Low
CP (Critical Parent)	leadership	fecklessness
NP (Nurturing Parent)	caring	cold-hearted
A (Adult)	calm	sentimental
FC (Free Child)	curiousness	impotence
AC (Adapted Child)	cooperativeness	contumacious

Table 2. Patterns of Egogram

Class	Sub Class	Pattern
High Type	CP high type	abbbb, acccc, etc.
	NP high type	babbb, caccc, etc.
	A high type	bbabb, ccacc, etc.
	FC high type	bbbab, cccac, etc.
	AC high type	bbbba, cccca, etc.
	P high type	aabcc, aaccc, etc.
	C high type	ccbaa, cccaa, etc.
Low Type	CP low type	caaaa, cbbbb, etc.
	NP low type	acaaa, bcbbb, etc.
	A low type	aacaa, abcba, bbcbb, etc.
	FC low type	aaaca, bbbcb, abbca etc.
	AC low type	aaaac, bbbbc
Mixed Type	trapezoid type I, II, III	caaac, caacc, ccaac
	U type I, II, III	accca, accaa, aacca
	N type I, II, III	cacba, cabca, cbaca
	inverted N type I, II, III	acabc, acbac, abcac
	M type	cacac
	W type	acaca
Flat Type	flat type I, II, III	aaaaa, bbbbb, ccccc

4. Proposed Method

4.1. *Framework of ego-state estimation*

We collected tweets that answered egogram using Twitter API. Sentence distributed expression vectors were then generated based on the tokenized tweets by using the Japanese morphological analysis engine MeCab[a].

To train the sentence-distributed expressions, a large amount of text data is required. Accordingly, tweet sentences were randomly and separately collected from the tweet data described above and used to train the distributed expression vector generation model. This is equivalent to pre-training in machine learning. (Hereafter, a sentence distributed expression vector will be referred to as a sentence vector.)

The tweet sentences of each user were converted into sentence vectors based on the sentence vector generation model. The egogram results (ego-state levels) for each user of the tweets were then used as a label. In this way, the sentence vectors and their labels were used as training data for machine learning. We created an ego-state level classifier for each ego-state based on the training data. Fig.1 shows the construction flow of the ego-state estimation model.

In our study, we exclusively used reliable tweets and accounts by manually checking the training data. The output flow of the classification result by the ego-state estimation model is shown in Fig. 2.

As each ego-state was expressed using one of three levels, we confirmed the effectiveness of the model by calculating how many of each of the estimated ego-state levels matched the answers. Combinations of ego-states can be classified into patterns based on the shape of the egogram. Based on this, we conducted an additional evaluation involving the accuracy of the pattern.

We also considered ego-state time-series variation, analyzing changes in the average match rates to determine how/whether the estimated ego-states would be different before and after the day the user received the egogram judgment.

4.2. *Tweet Collection*

We collected the tweet data to construct our ego-state estimation model by using Twitter API[22]. The following two conditions were set as conditions for the target accounts from which to collect tweets:

1. The user received an egogram judgment on the egogram assessment website [21] and reported a web address accessing the assessment results.
2. The account was not a “bot” account that automatically tweets several fixed phrases regularly.

We targeted user accounts that satisfied these two conditions. We then collected a maximum of 200 tweets posted for each account after posting the egogram assess-

[a] http://taku910.github.io/mecab/

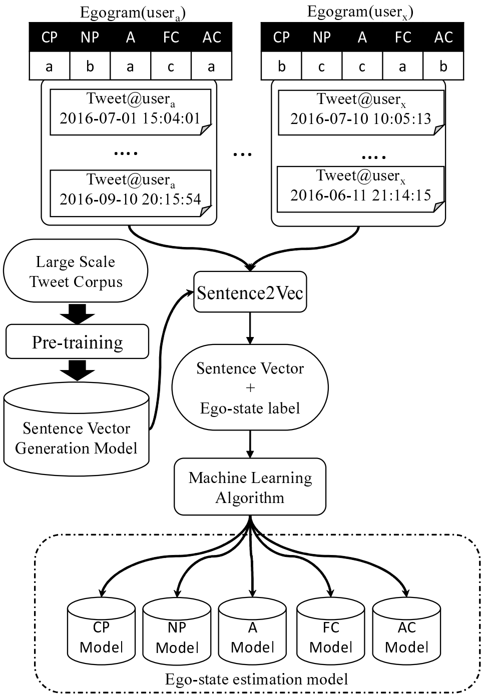

Egogram(user$_a$)
CP NP A FC AC
a b a c a
Egogram(user$_x$)
CP NP A FC AC
b c c a b
Tweet@user$_a$ 2016-07-01 15:04:01
....
Tweet@user$_a$ 2016-09-10 20:15:54
...
Tweet@user$_x$ 2016-07-10 10:05:13
....
Tweet@user$_x$ 2016-06-11 21:14:15
Large Scale Tweet Corpus
Pre-training
Sentence Vector Generation Model
Sentence2Vec
Sentence Vector + Ego-state label
Machine Learning Algorithm
CP Model
NP Model
A Model
FC Model
AC Model
Ego-state estimation model

Fig. 1. Flow of Training

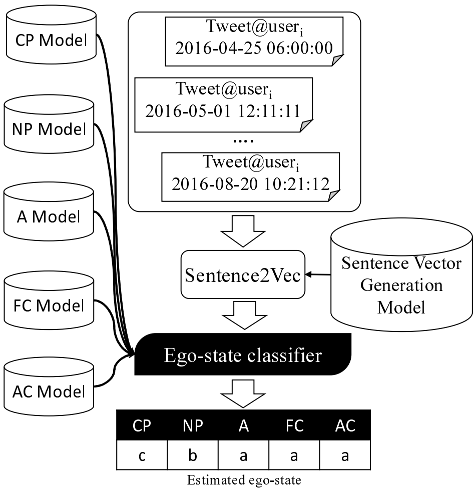

CP Model
NP Model
A Model
FC Model
AC Model
Tweet@user$_i$ 2016-04-25 06:00:00
Tweet@user$_i$ 2016-05-01 12:11:11
....
Tweet@user$_i$ 2016-08-20 10:21:12
Sentence2Vec
Sentence Vector Generation Model
Ego-state classifier
CP NP A FC AC
c b a a a
Estimated ego-state

Fig. 2. Flow of Estimation

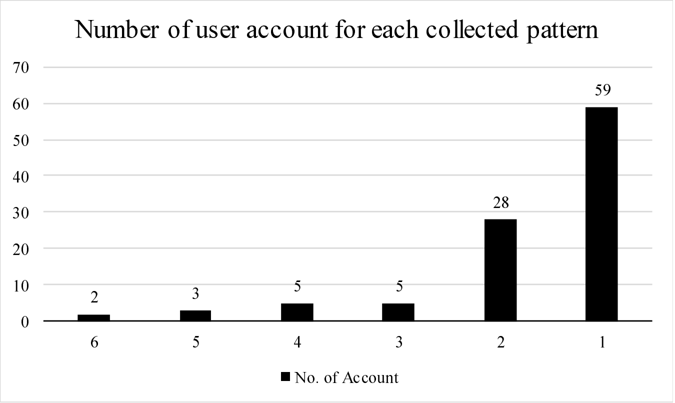


Fig. 3. The number of the patterns for each collected data

ment result. Fig. 3 shows the number of patterns for the collected data by histogram.

4.3. *Feature Extraction from Tweet*

We removed noise such as retweets or short URLs from the collected tweets. The noise-removed tweets were then converted to tokenized text by a morphological analyzer. The text data were converted into a sentence vector in tweet units using Sentence2Vec.

Sentence2Vec is a tool used to convert a sentence into a distributed expression vector. In our study, we used an approach based on Paragraph Vector, as proposed by Mikokov, et al.[b]. Implementation of this approach was based on Word2Vec and the Distributed Memory Model of Paragraph Vectors (PV-DM)[20]. For the task of Sentiment classification (5-class), Paragraph Vector showed similar or better performance than the existing method. As a result, to handle the task of sorting each ego-state into three classes, we believed that we could expect satisfactory performance using Paragraph Vector as the feature quantity.

We chose not to adopt an approach extracting features of limited parts of speech. At present, it is not confirmed what words or phrases affect the classification of personality patterns. If a part of speech of a word that does not make sense by itself is used as a feature, it may adversely affect the precision of the classifier. As a sentence makes sense by lining up a series of words, using only specific words as a feature may well prove unfruitful. Therefore, we decided to use the distributed representation of a sentence (sentence vector) as the feature because it can use every word included in the training data as a feature, as well handle the feature quantity with fixed dimensionality.

4.4. *Degree of Sentence Formality*

In a previous study, spoken language and written language were classified using the word feature. We have attempted to investigate the word feature for both formal and informal writing styles for personality estimation.

In general, formal words tend not to appear often in spoken sentences, where informal words are far more common. “Nihongo-no Goitokusei: Lexical Properties of Japanese”[18] introduces the term “word familiarity degree,” whereby the familiarity of words is represented by a numerical value. It is thought that this familiarity degree can be effective as a clue in estimating the formality of a sentence. In “Gokan no jiten,” Nakamura[19] classified colloquial and written language. We believe that this classification is effective as a feature of a sentence’s formality degree.

Word familiarity degree assigns a value in the range of 1 to 7 to words—the larger the value, the more familiar the words, Highly-scored words are widely known to society at large. In this paper, we classified word familiarity degree into 7 levels and used this as a feature.

We used a dictionary in which the registered words were classified into three categories: formal/informal/normal. For words included in a given sentence, we

[b] https://github.com/klb3713/sentence2vec

referred to this dictionary and used appearance frequency for each category as a feature. We prepared a dataset consisting of 10,000 tweets that included slang and 10,000 tweets that included only standard words for a binary classification of formal/informal.

We created a binary classification model by extracting a 10-dimension feature vector as described above from each tweet in the dataset and trained the dataset with a support vector machine. We confirmed the model's accuracy using a five-fold cross validation. As a result, a score of approximately 0.855 was obtained. It was concluded, then, that this feature was effective in judging writing style.

5. Experiment

To establish the effectiveness of the proposed method, we conducted an ego-state level estimation experiment under various conditions and employed a cross-validation test using the training data and an open test using the users' other accounts that were not included in the training data. We next attempted to change the machine learning algorithm, the parameter setting of Sentence2Vec, and the kind of the training corpus for creating the sentence vector, and compared these conditions.

We then compared performance using a bag-of-words-based method as our baseline. The experimental process and experimental conditions are described in detail in the following section.

5.1. *Experimental Conditions*

The flow of the experiment is shown below:

(1) We create the training data. The data consist of a text corpus randomly collected from Twitter and tokenized by the morphological analyzer MeCab. We then create a sentence vector generation model from the data using Sentence2Vec.
(2) The tweet corpus with annotated ego-state labels is tokenized, and the corpus is converted into sentence vectors by unit of tweet using the sentence vector generation model created in (1)
(3) We construct the ego-state level estimation model for each type of ego-state by coordinating ego-state in each pattern of the egogram.
(4) We evaluate the model by using a cross-validation test and an open test

To avoid learning bias, we used only data randomly selected for one user for each personality pattern as training data. We then used nine different algorithms that can accommodate multivalued classification as our machine learning method for generating the ego-state estimation model. Table 3 shows the algorithms and their abbreviations.

A DN (Deep Neural Network) is a feedforward neural network that consists of three hidden layers. In this study, the number of nodes in each layer was set at

$hu_1 = 20, hu_2 = 10, hu_3 = 20$. The nodes between each of the layers are fully-connected and feedforwarded. Fig. 4 indicates the structure of the neural network that was used.

We conducted an experiment to evaluate the accuracy of the model using a 10-fold cross-validation, coordinating the training data that did not include the same user or personality pattern as the users included in the test data. We then evaluated the ego-state level estimation model by using the tweet data of the other 59 user accounts as test data that were different from the training data. The calculation of estimation accuracy ($Match$) as an evaluation basis is shown in Eq. 1. Here, L^i_{pred}, L^i_{obs} indicate the estimated label of ego-state i and the correct label's level value. The function $mc()$ in Eq. 1 and Eq.2 is a function that takes a value of 1 when the levels are the same for ; otherwise it takes a value of 0.

Table 3. Machine learning algorithms used for ego-state level estimation

Training algorithm	Abbrev.
AdaBoost	AB
Extra Trees	ET
Gradient Boosting	GB
Random Forest	RF
LBFGS Logistic Regression	LB
Pegasos Logistic Regression	PL
Averaged Perceptron	AP
Truncated Gradient Logistic Regression	TL
Deep Neural Network	DN

$$mc(L^i_{pred}, L^i_{obs}) = \begin{cases} 1 \ (L^i_{pred} = L^i_{obs}) \\ 0 \ (otherwise) \end{cases} \tag{1}$$

$$Match = \frac{\sum_{i=1}^{n} mc(L^i_{pred}, L^i_{obs})}{n} \tag{2}$$

5.2. *Result*

Fig. 5 shows the match rate for each classifier in the cross-validation test. According to the results, the proposed method (DN) had greater accuracy than the baseline method.

Fig. 6 shows the results of the open experiment, where, again, the proposed method (DN) has the advantage. We conducted the additional ego-state estimation experiment with the open data using the DN ($win = 10, dim = 300, EM$) that achieved the best accuracy in the open experiment, using the sequential 5, 10, 15, ..., 50 tweets as training data and test data. Fig. 7 shows the result. When the

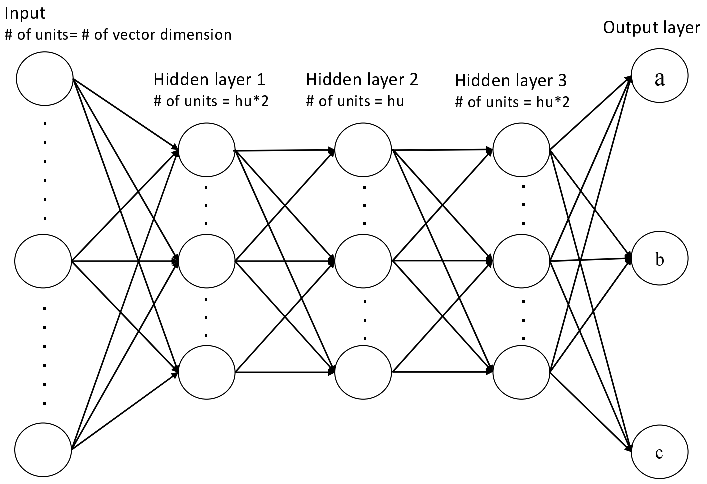


Fig. 4. Structure of Deep Neural Network

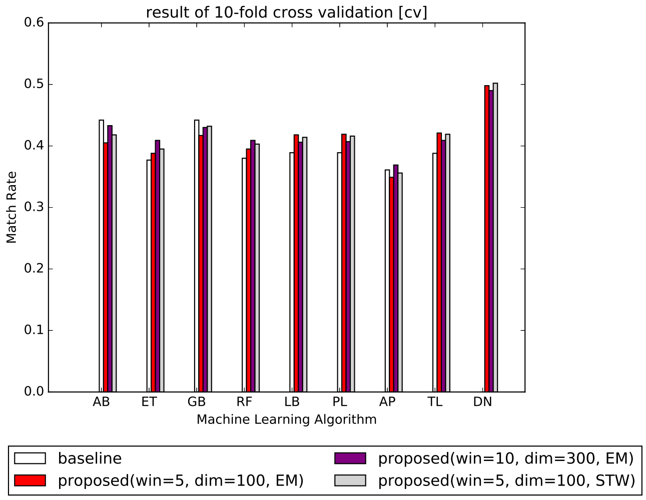


Fig. 5. Result of cross validation test

number of hidden layer nodes was set as 100, 50, 100 and the batch size was 128, and the averaged sentence vector of 35 tweets was used, the highest average match rate, 0.488, was produced.

It was observed that as the number of sequential tweets approaches and exceeds 40, the match rate tends to decrease. Consequently, it may be sufficient to use 10 to 35 tweets when creating the averaged sentence vector. For a 10-tweet averaged sentence vector, the accuracy for each ego-state is shown in Table 4. As can be seen,

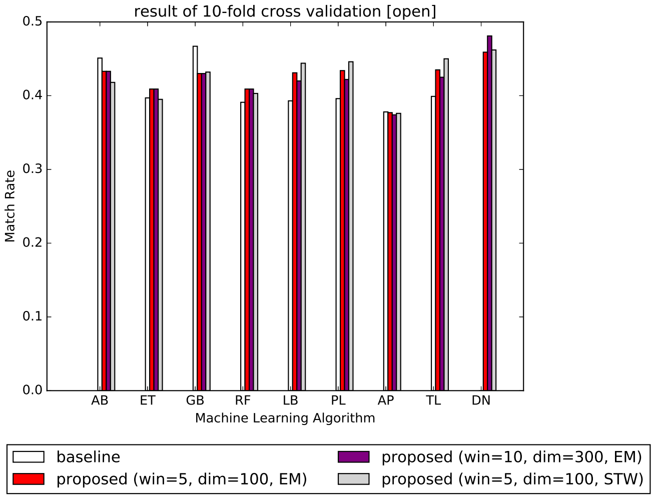


Fig. 6. Result of open test

the accuracy for CP and A are high, while the accuracy for NP is comparatively low. This was not the same for other parameters or other algorithms.

Fig. 8 shows the average match value for each personality pattern in the open experiment. As shown, the match rate of a personality pattern that includes an ego-state level of ‘c’ tends to decrease.

Table 4. Accuracy of each ego state: DN ($win = 10, dim = 300, twn = 10, hu = 10, mb = 128, epoch = 500$)

Ego state	Accuracy
CP	0.561
NP	0.325
A	0.561
FC	0.516
AC	0.446

5.3. *Analysis*

We investigated the change in estimation results in a time series. We collected a number of weeks of tweets for user accounts after 31 Aug. 2016. Any noise in the tweets was automatically removed. We calculated the match rate with the correct egogram using training and classifying sentence vectors based on DN generated from tweets as the feature.

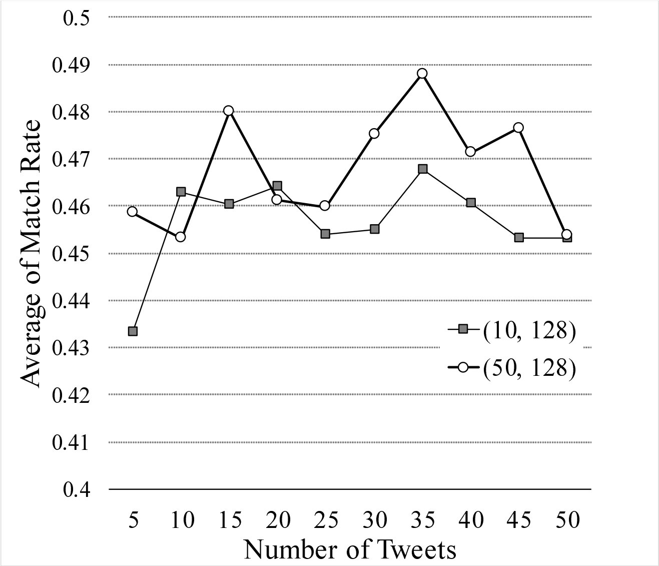


Fig. 7. Match Rate transition by number of tweets

Fig. 9 shows the results of the analysis. The horizontal axis indicates the elapsed days from the date of egogram assessment; the vertical axis indicates the average match rate. The minus value on the horizontal axis indicates the average value of the match rate of the ego-state estimation before the egogram assessment. From this result, it was found that the match rates for the periods after the egogram assessment were lower than for those before the assessment. Because a user behaves (and writes) based on his/her self-personality, it was thought that after the egogram assessment the content of the user ' s tweets would change. However, the match rates vary, as did the number of tweets and the differences for each user.

In an additional experiment, we added a formal/informal feature to the sentence vector and used an ego-state estimation model based on the modified vector. We conducted an experiment using this model and open data, and compared the accuracy of the estimated personality pattern to the model based only on the sentence vector. Fig. 10 shows the comparison results. The combinations of values (10, 256), (50, 512), (100, 1024) shown at the bottom of the figure indicate the number of hidden layer nodes (hu) and mini batch sizes (i.e., the number of examples) used in each learning step. The number of hidden layer nodes is set as $hu_1 = 2 * hu, hu_2 = hu, hu_3 = 2 * hu$. We then set the learning steps as 2000 for each condition. It was found that accuracy tends to be high when the additional feature is used and that estimation accuracy was related to degree of formality. The highest accuracy was obtained when, in training, the number of hidden layer nodes is $hu_1 = 200, hu_2 = 100, hu_3 = 200$ and the mini batch size is 256.

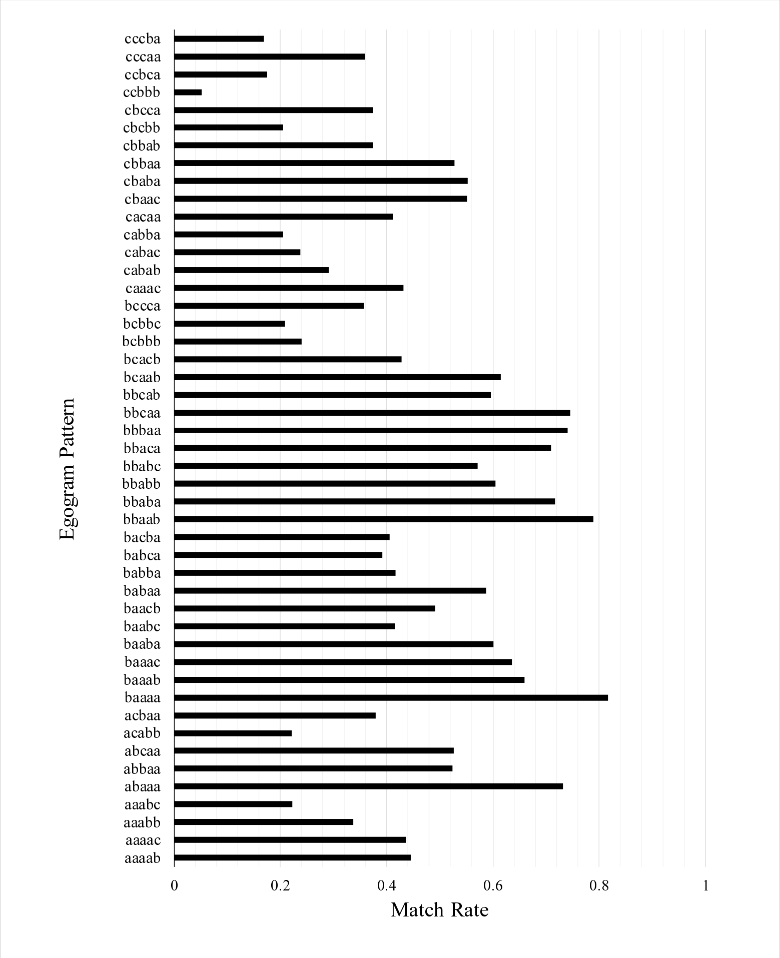


Fig. 8. Match Rate of each egogram pattern

6. Conclusion

We were able to achieve a higher match rate than the baseline method in the ego-state estimation task by combining the sentence distributed expression and deep learning. We believe that using sequential tweets increases information. By using the averaged vectors of 35 sequential tweets, we produced the best match rate. However, it seems clear that personality is not strongly expressed in all tweets. Therefore, a larger dataset is likely needed for training when using an integrated vector from sequential tweets.

In our approach, we applied a method that trains a sentence vector from a large corpus as pre-training. However, it is possible to train raw sentences using a neural network such as a Recurrent Neural Network (RNN) or Long Short Term Memory

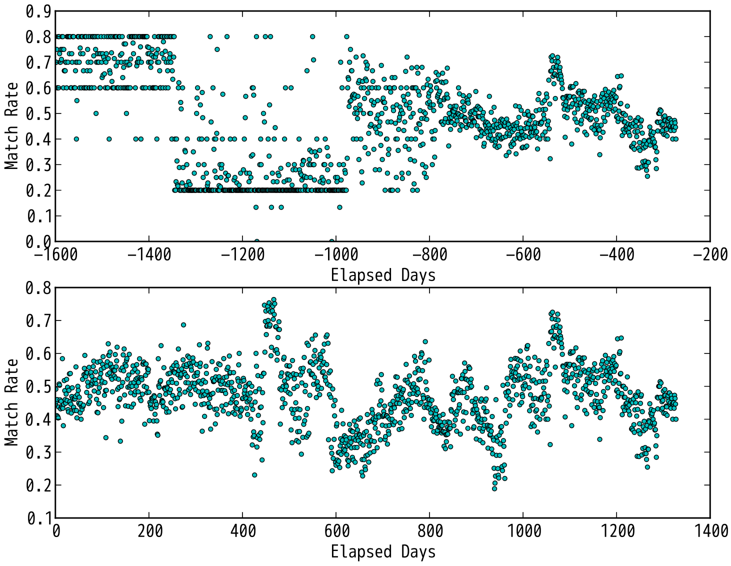
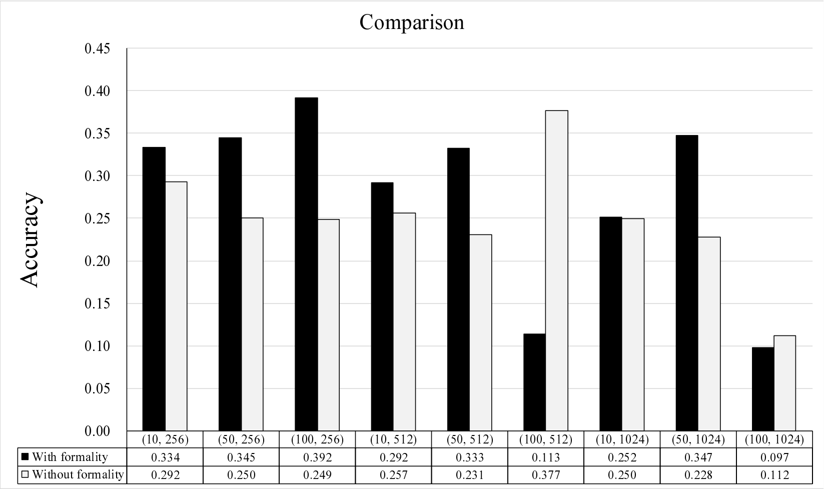


Fig. 9. Match Rate for each elapsed days



	(10, 256)	(50, 256)	(100, 256)	(10, 512)	(50, 512)	(100, 512)	(10, 1024)	(50, 1024)	(100, 1024)
■ With formality	0.334	0.345	0.392	0.292	0.333	0.113	0.252	0.347	0.097
□ Without formality	0.292	0.250	0.249	0.257	0.231	0.377	0.250	0.228	0.112

Fig. 10. Comparison between with formality feature and without formality feature

(LSTM) if we have a sufficiently large labelled dataset.

We also investigated ego-state changes over time and found that the match rate for the time after assessment is lower than for before assessment. However, we did not confirm how/whether the user changes before and after his/her egogram assessment. This would require a topic change in tweet contents since the contents might change before/after an event that would be a factor in changing the personality of the user.

Moreover, it was found that the accuracy of our ego-state estimation changed (improved) when a feature expressing the degree of formality of the tweet was added.

In confirming the match rate of the combination pattern of ego-state levels, it was found that the accuracy for patterns that include a low ego-state level tends to relatively low. As a result, it is thought that it is difficult to appeared that lowness of the ego-state level on the surface such as contents of tweet. In the future, we plan to collect more personality patterns among the 243 possibilities. We will then add users associated with each pattern, conduct evaluation experiments, and perform our analysis in greater detail. Finally, we intend to consider the application potential of information recommendations by exploring the relationship between a user's personality and time, season, etc., based on personality estimation in a time series.

Acknowledgments

This research has been partially supported by JSPS KAKENHI Grant Number 15H01712, 15K16077, 15K00425.

References

1. Alastair J. Gill, Jon Oberlander and Elizabeth Austin, "Rating e-mail personality at zero acquaintance," Vol. 40, No. 3, pp. 497–507, 2006.
2. Jennifer Golbeck, Cristina Robles, Michon Edmondson and Karen Turner, "Predicting Personality from Twitter," In Proc: 2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing, pp. 149–156, 2011.
3. Rong Hu and Pearl Pu, "Enhancing collaborative filtering systems with personality information," In: Proceedings of the ACM RecSys, pp. 197–204, 2011.
4. Fabio Celli, Fabio Pianesi, David Stillwell and Michal Kosinski, "Workshop on Computational Personality Recognition: Shared Task," AAAI Technical Report WS-13-01, Computational Personality Recognition(Shared Task), 2013.
5. Golnoosh Farnadi, Susana Zoghbi, Marie-Francine Moens, and Martine De Cock, "Recognising Personality Traits Using Facebook Status Updates," AAAI Technical Report WS-13-01, Computational Personality Recognition(Shared Task), 2013.
6. Fabio Celli, Sergio Davalos, Marie-Francine Moens and Martine De Cock, "Computational personality recognition in social media," User Modeling and User-Adapted Interaction, Vol.26, Issue 2, pp.109–142, 2016.
7. Noriyuki Okumura and Manabu Okumura: "A Construction of Knowledge Base for Personality Estimation based on Submitted Text Data in Twitter or Blogs," In Proc: KEOD2015, pp.418–423, 2015.
8. Atsunori Minamikawwa and Hiroyuki Yokoyama: "Blog tells what kind of personality you have: egogram estimation from Japanese weblog, " In Proc: CSCW '11 Proceedings of the ACM 2011 conference on Computer supported cooperative work, pp. 217–220, 2011.
9. Atsunori Minamikawwa and Hiroyuki Yokoyama: "Personality Estimation Based on Weblog Text Classification, " Volume 6704 of the series Lecture Notes in Computer Science, Vol.6704, pp. 89–97, 2011.

10. Satoshi Tanaka, Kazuyuki Matsumoto, Minoru Yoshida and Kenji Kita: "Personality Analysis Method of Twitter User for Information Recommendation, " In Proc: The 30th Annual Conference of the Japanese Society for Artificial Intelligence, 2016, 3B4-NFC-05b-3, 2016.
11. Tetsuya Nasukawa, Koichi Kamijo, Masahiro Yamamoto and Hideya Kitamura: "Survey on Linguistic Feature for Personality Estimaton of Writer, " In Proc: the 22th annual meeting of the Association for Natural Language Processing, pp. 1181–1184, 2016.
12. IBM Watson, Personality Insights: `https://www.ibm.com/watson/developercloud/personality-insights.html`.
13. Ayako Baba, Mathieu Bertin, and Yasuo Tanida: "Developing consumer's value structure model and type "Societas" as social knowledge, " The 27th Annual Conference of the Japanese Society for Artificial Intelligence, pp. 1–4, 2013.
14. Lewis R. Goldberg: "An Alternative "Description of Personality": The Big-Five Factor Structure, " Journal of Personality and Social Psychologs, Vol. 59, No.6, pp. 1216–1229, 1990.
15. John M. Dusay, "Egograms," Bantam USA, 1980.
16. Tokyo University Egogram New Ver. II, Kaneko Publisher, 2006.
17. Takako Saito, Tomoyasu Nakamura, and Madoka Yokoyama: "Standardization of Big Five scales using the Adjective Check List, " Kyushu University Psychological Research, Vol.2, pp. 135–144, 2001.
18. N. Amano, K. Kondo, S. Sakamoto and Y. Suzuki, NTT psycho-linguistic databases "Lexical Properties of Japanese," 1999, San-seido, 2011.
19. Akira Nakamura, Japanese Gokan no jiten (in Japanese), Iwanami Publisher.
20. Quoc Le, and Tomas Mikolov: "Distributed Representations of Sentences and Documents, " In Proc: the 31st International Conference on Machine Learning (ICML2014), 2014.
21. Personality Assessment by Egogram: `http://www.egogram-f.jp/seikaku`.
22. Twitter API: `https://dev.twitter.com/`.

Kazuyuki Matsumoto

He received the PhD degree in 2008 from Tokushima University. He is currently an assistant professor of Tokushima University. His research interests include Affective Computing, Emotion Recognition, Artificial Intelligence and Natural Language Processing. He is a member of IPSJ, ANLP, IEICE and IEEJ.

Satoshi Tanaka

He received the BS in 2016 from Tokushima University. He is currently a master course student in Tokushima University. His research insterests include Personalitiy Analysis, Artificial Intelligence.

Minoru Yoshida

He is a lecturer at the Department of Information Science and Intelligent Systems, University of Tokushima. After receiving his BSc, MSc, and PhD degrees from the University of Tokyo in 1998, 2000, and 2003, respectively, he worked as an assistant professor at the Information Technology Center, University of Tokyo. His current research interests include Web Document Analysis and Text Mining for the Documents on the WWW.

Kenji Kita

He received the B.S. degree in mathematics and the PhD degree in electrical engineering, both from Waseda University, Tokyo, Japan, in 1981 and 1992, respectively. From 1983 to 1987, he worked for the Oki Electric Industry Co. Ltd., Tokyo, Japan. From 1987 to 1992, he was a researcher at ATR Interpreting Telephony Research Laboratories, Kyoto, Japan. Since 1992, he has been with Tokushima University, Tokushima, Japan, where he is currently a Professor at Faculty of Engineering. His current research interests include multimedia information retrieval, natural language processing, and speech recognition.

Fuji Ren

He received his B.E. and M.E. degrees from Beijing University of Posts and Telecommunications, Beijing, China, in 1982 and 1985, respectively. He received his Ph.D. degree in 1991 from Hokkaido University, Japan. From 1991, he worked at CSK, Japan, where he was a chief researcher of NLP. From 1994 to 2000, he was an associate professor in the Faculty of Information Sciences, Hiroshima City University. He became a professor in the Faculty of Engineering of the University of Tokushima in 2001. His research interests include natural language processing, artificial intelligence, language understanding and communication, and affective computing. He is a member of IEICE, CAAI, IEEJ, IPSJ, JSAI, and AAMT, and a senior member of IEEE. He is a fellow of the Japan Federation of Engineering Societies. He is the president of the International Advanced Information Institute.