IEEJ TRANSACTIONS ON ELECTRICAL AND ELECTRONIC ENGINEERING
IEEJ Trans 2018; **13**: 1446–1453
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI:10.1002/tee.22711

Paper

Sentiment Analysis Method Based on an Improved Modifying-Matrix Language Model

Zhao Han[*,**], Non-member
Fuji Ren[*a], Member
Duoqian Miao[**], Non-member

In this paper, a sentence-level sentiment analysis method is proposed to deal with sentiment measurement and classification problems. It is developed from a model called the synthetic and computational language model (SCLM), which represents modifying and modified information, respectively, using matrices and vectors. In the proposed method, a global modifying matrix of a sentence is constructed, the determinant value of this matrix is calculated and adjusted, and then the final value is used as the sentiment value of the sentence. Regression experiment shows that the deviation between the output sentiment and the target sentiment does not exceed a class distance of five classes. The classification experiment shows that the proposed method has improved most of the performance compared to the simplified SCLM and in some cases, such as in 'very positive' class and 'very negative' class, reaches higher precision performance than the baseline method.

Keywords: affective computing; sentiment analysis; emotion classification; language model; neural network

Received 1 October 2016; Revised 24 January 2018

1. Introduction

Sentiment analysis (SA) is one of the most important research topics in information processing, and within SA, research on sentiment measurement and classification has been a very popular topic [1]. The main task in sentiment measurement and classification is to classify the text paragraph into different sentiment classes. Such technology has been applied in many fields, for example, opinion trend tracking of Twitter topics [2], customer review mining in marketing [3], and affective interaction between human and a dialogue robot [4,5]. Most of the research is conducted at one of the three levels: document level, sentence level, or attribute level [3], and at each level it involves different technologies at language representation methods and machine learning methods [6]. For some applications such as social robots [4] and Twitter mining [2], the text length is usually very short and the SA is at the sentence level most of the time.

Here we focus on two problems in short text analysis. (Here, a 'short text' is defined as a single sentence ending with a period. It is different from the long paragraph with more than two sentences and sentence relation within the paragraph.) One is that, for social dialogue or Twitter, flexible change of the spoken language and the Internet language is very common. Different from a static document text, such as a novel or a report, the new language information can be dynamically added into the existing context, such as by sending comments to a twitter or adding further explanations during a human–robot interaction. Another is that, based on the running environment and system resource limits, the system must allow the adjustment of the fineness of the semantic analysis and affective computing. For example, during a human–robot interaction, sometimes the robot only needs to classify the user's input into three sentiment classes: *positive*, *neutral*, and *negative*, while sometimes it also needs to discriminate the sentiment changing extent between two sentences of the same class, such as from *common positive* to *extreme positive*. To solve these problems, a language representation model named synthetic and computational language model (SCLM) was proposed by Han *et al.* [7], which represents modifying and modified information using matrices and vectors, respectively. In SCLM, the modifying matrices contain more affective information than the vectors, while the modified vectors contain more semantic information than the matrices. The determinant value of the matrix will be the sentiment tendency value of a modifying word or a sentence part.

One of the advantages of SCLM is that all the transformation or modification can be represented by adding multiplying matrices flexibly, and can increase or decrease the computing fineness according to the operating environment through many ways such as changing the dimension of the matrices. Another advantage is that it treats all the sentiment information as 'modifying', including 'negative modifying', so it is easy to deal with all kinds of sentences, irrespective of whether they are positive type or negative type. However, the SA performance of the original SCLM was unsatisfactory. Although SCLM was good at discriminating the sentiment changing such as from '*I am happy*' to '*I am very happy*', the test result on a large sentiment corpus such as the Stanford Treebank Dataset [8] still needed improvements. To improve the SA performance of SCLM, a new SA method is proposed in this paper. In the proposed method, a global modifying matrix of a sentence is constructed, the determinant value of this matrix is calculated and adjusted, and then the final value is used as the sentiment value of the sentence.

The rest of the paper is organized as follows: Section 2 introduces related work on language models and machine learning methods for SA. Section 3 introduces SCLM and the proposed method. Section 4 describes the experiment details and presents the experimental results and analyses. Section 5 comprises the conclusion and future work.

[a] Correspondence to: Fuji Ren. E-mail: ren@is.tokushima-u.ac.jp

[*] Faculty of Engineering, Tokushima University, Tokushima 770-8506, Japan

[**] The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China

2. Related Work

2.1. Language representation for sentiment analysis

The first step of SA is to change the input text into basic information representation such as tokens [9], POS (part of speech) tags [10], and parsing dependencies [11]. For analysis at the sentence level, POS tags and parsing dependencies are very useful since they contain much information about sentiment word positions and modifying targets [12]. Many language models focus on this kind of features using various data formats, e.g. knowledgebase [13,14] (such as HowNet [15] and WordNet [16]), and corpus [17,18]. The statistical information [19,20] or other mathematical information [21] of features is also widely used in sentence preprocessing. For oral language text or network parlance processing, other features such as using environment and text source also need to be considered, e.g. Twitter hashtags and smileys [22] and user behavior [23].

Another aspect that may influence the sentiment tendency is the topic and opinion of the sentence. For example, during a dialogue, the topic of celebrating a festival is mostly in the positive sentiment class while the topic of earthquake is mostly in the negative class. Many language models for topic representation of short text have been proposed in recent years such as the Biterm Topic Model [24], and some technologies for long text topic mining have also been developed into technologies for short text topic mining [25]. Some methods for predicting user opinions can also be used in short text interaction [2,26].

There is also research work that focuses on sentiment processing of negative text. Several methods have been proposed to detect the negative emotion and sentiment among large scale of data, such as news [27,28] and posts on Facebook [29,30]. Some methods for speech negative emotion have also discussed the text features of negation [31,32].

Since sentiment information usually depends on both semantic information and affective words, for a synthetic requirement, a language model called SCLM [7] which can represent both semantic and sentiment information is first proposed for social robots. SCLM has some similarities with NaturalLI [33] in the idea of linguistic computing, and the matrix representation methods of the proposed language model resemble recursive matrix–vector spaces [34] and other neural probabilistic language models [35,36]. The difference is that in SCLM the elements of the matrices and vectors are based on the sentence or paragraph level with speaker source and perspective coordinate information, while the others are based on word level and usually the perspective information is ignored at the model representation. For our dialogue system, SCLM can fit the needs much better.

2.2. Supervised machine learning and neural networks

Supervised machine learning is a kind of machine learning where instances are given with known labels or the corresponding correct outputs [37]. Of all the supervised learning methods, supervised neural networks (NNs) model [38–40] has become popular in recent years because of the development of deep learning (DL) [41–43], and back-propagation neural network (BPNN) is one of the most common supervised training algorithms [44,45]. DL can be treated as an improved and much more complex version of NN, and it often deals with more complex information coding and encoding. In the area of DL, the standard recursive neural network (RNN) [46] model is the simplest NN-based model, and based on RNN, matrix–vector RNN (MV-RNN) [34] and recursive neural tensor network (RNTN) [47] have been developed. These three models have achieved good performance in language sentiment measurement. Other recent research on NN and DL has also made very significant achievements in artificial intelligence and pattern recognition [48], but they still have a lot of research space.

The advantages of the NN method over traditional classifiers are its nonparametric nature, arbitrary decision boundary capabilities, easy adaptation to different types of data and input structures, fuzzy output values that can enhance classification, and good generalization for use with multiple images [49]. Considering the time and memory cost of training deep networks by the DL method, in this paper we only use the traditional BPNN methods to train our system in the first step of our research.

3. Proposed Method

3.1. Sentence representation and sentiment measurement of SCLM

Using SCLM, a single declarative sentence can be represented by

$$S = \begin{cases} T_0, & n = 0 \\ T_0 + \sum_{i=1}^{n} C_i, & n = 1, 2, 3, \ldots \end{cases} \tag{1}$$

where T_0 represents the main trunk clause of the sentence, and C_i represents the subordinate clauses. n stands for the number of subordinate clauses.

For T_0 and each C_i, we use (2) to represent

$$\left(\prod_{\mathrm{j}=0}^{m} M_{kj} \right) V_k, \qquad k, \mathrm{j}, m = 0, 1, 2, \ldots \tag{2}$$

where M_{kj} or V_k is the smallest unit of the model representation. When $k = 0$, V_k represents the main trunk T_0, and when k is any other integer, V_k represents the clause C_i. For each V_k, the elements in it represent the modified words in the sentence, and the elements in M_{kj} must correspond to those in the modified vector. Formula (3) with () is an example of using a vector of four dimensions, and the modifying matrix is of 4×4 size:

$$V_k = \begin{bmatrix} \text{subject} \\ \text{predicate verb/copula} \\ \text{direct object} \\ \text{indirect object} \end{bmatrix} \tag{3}$$

$$M_{kj} = \begin{bmatrix} \mathrm{DM_{subj}} & s-p & s-d & s-i \\ p-s & \mathrm{DM_{prev}} & p-d & p-i \\ d-s & d-p & \mathrm{DM_{dobj}} & d-i \\ i-s & i-p & i-d & \mathrm{DM_{iobj}} \end{bmatrix} = \begin{bmatrix} \mathrm{row_{subj}} \\ \mathrm{row_{prev}} \\ \mathrm{row_{dobj}} \\ \mathrm{row_{iobj}} \end{bmatrix} \tag{4}$$

In M_{kj}, 'subj' means 'subject', 'prev' means 'predicate verb or copula', 'dobj' means 'direct object', and 'iobj' means 'indirect object'. The elements on the matrix diagonal such as $\mathrm{DM_{subj}}$ are the most direct elements modifying the vectors (directly modifying). $\mathrm{row_{subj}}$, $\mathrm{row_{prev}}$, $\mathrm{row_{dobj}}$, and $\mathrm{row_{iobj}}$ represent the rows of the matrix. Elements at the other positions show the hidden relations between the indirect modifying words and the modified words. Because of the multiplying rules between a matrix and a vector, the element 'A–B', which is different from the element 'B–A', has a directional meaning from A to B. For example, 's–p' means the effects from the 'subject' to the 'predicate verb/copula', and will function on the 'predicate verb/copula' part of the vector after the multiplying process, while 'p–s' means the effects from the 'predicate verb/copula' to the 'subject' and will function on the 'subject'. If the element in the position of 'A–B' is empty, it means that there is no modifying relationship from 'A' to 'B'.

For each M_{kj}, $|M_{kj}|$, which represents the determinant value of M_{kj}, can be used as a kind of sentiment value in the sentence. The determinant can be calculated using the determinant calculation method in linear algebra [50,51]. Since the matrix rows have corresponding modifying relationships with the vector elements, we can use the vector like the one described in (3) to represent the selected syntax features. So we can say the selected syntax features of the given four dimensions SCLM are ‘subject’, ‘predicate verb/copula’, ‘direct object’, and ‘indirect object’. If we get all the syntax features, the elements not on the diagonal can be deduced by the diagonal elements.

Here is a simplified example of decoding a sentence: ‘*The film is painfully authentic, and the performances of the young players are utterly convincing*’, using a vector of four dimensions and matrices of 4×4 size (for a clear formula expression, we omit the elements that are not in the matrix diagonal, and use ‘$\square$’ to represent an empty position of the diagonal), like in (5). This sentence is selected from the Stanford Treebank Dataset [8]:

$$
\begin{aligned}
S &= T_0 + C_1 \\
&= (M_{00})V_0 + \left(\prod_{j=0}^{1} M_{1j}\right) V_1 \\
&= \begin{bmatrix} \text{The painfully authentic} & & & \\ & \text{(is)} & & \\ & & \square & \\ & & & \square \end{bmatrix} \begin{bmatrix} \text{film} \\ \text{be} \\ \square \\ \square \end{bmatrix} \\
&\quad + O(\text{and}) \begin{bmatrix} \text{the, } (s)\text{, of the young players, utterly convincing} & & & \\ & \text{(are)} & & \\ & & \square & \\ & & & \square \end{bmatrix} \begin{bmatrix} \text{performance} \\ \text{be} \\ \square \\ \square \end{bmatrix}
\end{aligned} \tag{5}
$$

In (5), T_0 represents the main trunk ‘The film is painfully authentic’ and C_1 represents the clause ‘and the performances of the young players are utterly convincing’. Since the word ‘and’ modifies the whole clause C_1 and in fact the words like ‘and’ or ‘but’ have operating meaning in SCLM (e.g. ‘and’ has the operating meaning of ‘plus’ while ‘but’ has the operating meaning of ‘minus’), we use $O(\text{and})$ to represent a matrix that modifies the overall elements of the trunk or clauses and has operating meaning.

3.2. Sentiment measurement in proposed method

The idealized SCLM is very difficult to realize. One of the problems is that there is still no effective definition for the operation of SCLM, such as the matrix multiplication in C_1 of (5). Another problem is that there is still no effective definition for us to get the modifying value of the hidden relation. For example, in the sentence ‘The performances of the young players are utterly convincing’, we can get the modifying relation from ‘utterly’ to ‘convincing’, visually or based on the parsing tree [8], while it is difficult for us to get the modifying value from ‘utterly’ to ‘performance’. In addition, SCLM depends too much on the dependency parsing results, but till now there is no effective tool for parsing a sentence with complex semantic dependency relations into SCLM representation. So in practical applications we usually use a simplified SCLM model to measure the sentiment of a sentence. We construct only one matrix and use the determinant value of this matrix as the sentiment value of the sentence. All the elements of the same parsing dependencies will be abstracted, and the average value of the sentiment tendencies will be used to calculate the element value in the modifying matrices. For example, in a sentence, the words that are modifying the direct objects, both in main trunks and subordinate clauses, will be abstracted and the average of their sentiment values will be used.

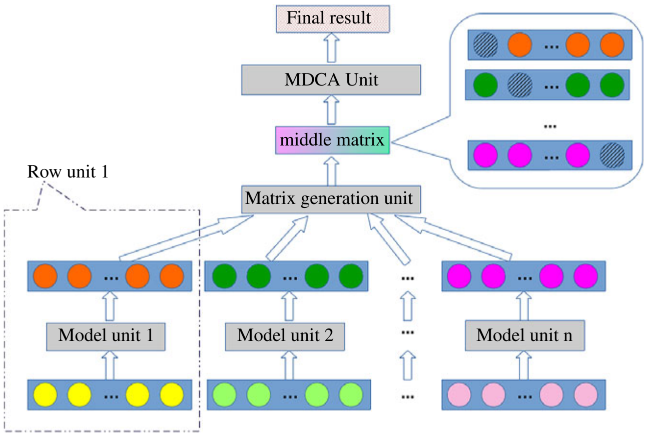


Fig. 1. Framework of sentiment matrix constructor

Then for one sentence, no matter how many clauses it has, we just need to calculate the determinant value of only one matrix. The matrix considers only the modifying words that directly modify the vector elements and uses only the sentiment value on its diagonal.

The sentiment measurement of the simplified SCLM is very good at discriminating the sentiment change of two similar sentences, but the SA performance of SCLM is unsatisfactory. To solve this problem, we develop a sentiment model based on the SCLM, and focus on the task of SA. A sentiment matrix constructor in n dimensions is mainly made up by n row units, a matrix generation unit, and a matrix determinant calculation and adjustment (MDCA) unit, as illustrated in Fig. 1. A row unit example is illustrated in Fig. 2, with the NN model inside used as the model unit (surrounded by a large gray rectangle), which is composed of several linear layers as hidden layers and a nonlinear layer as the output layer. The model unit receives the input values of the row unit and, after the model function, it will transmit the output values. A set of output values from one row unit will be passed to the matrix generation unit as one row of the matrix. The matrix generation unit will construct a sentiment matrix M, and add a model bias b on the diagonal elements of M to get a new matrix M' (details about b will be introduced in Section 3.3). The determinant value of M' will be calculated and adjusted by the MDCA unit, finally giving the sentiment value of the sentence.

If we choose n syntax features of the sentence, the modifying matrix M of the proposed method will be represented by

$$
M = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & m_{ij} & \dots \\ \dots & \dots & \dots & \dots \\ m_{n1} & m_{n2} & \dots & m_{nn} \end{bmatrix}
= \begin{bmatrix} \text{row}_1 \\ \text{row}_2 \\ \dots \\ \text{row}_i \\ \dots \\ \text{row}_n \end{bmatrix}
= \begin{bmatrix} f_1(F_1) \\ f_2(F_2) \\ \dots \\ f_i(F_i) \\ \dots \\ f_n(F_n) \end{bmatrix}, \qquad (i, \text{j} = 1, 2, \dots, n) \tag{6}
$$

In (6), m_{ij} is the matrix element in the row i and column j, row_i represent all the elements on the row i, f_i is the model function

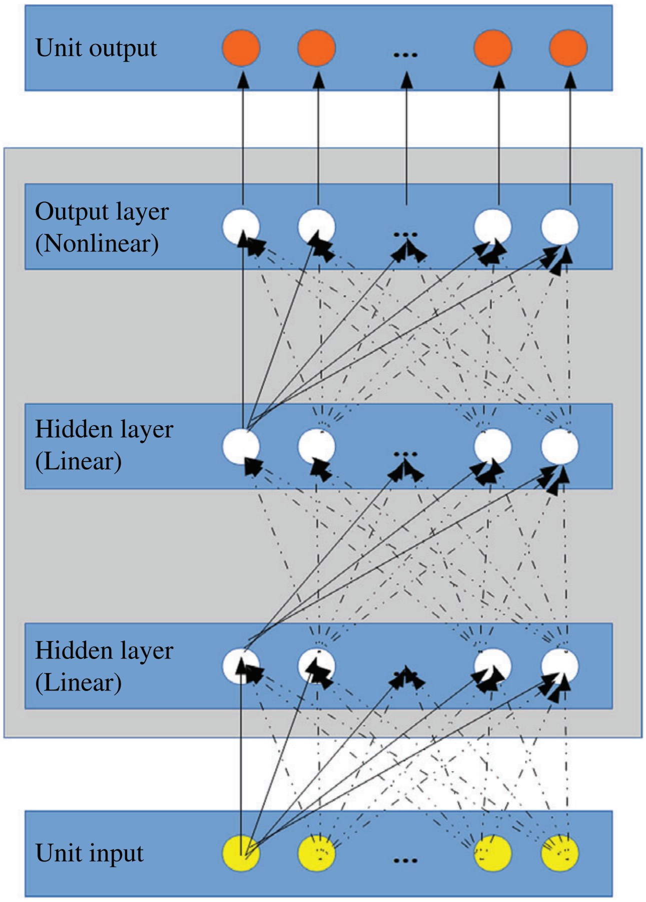


Fig. 2. Row unit with a model unit inside

of the *Model Unit* corresponding the row i, and F_i is the set of feature values of the row i. The feature values are based on the feature words, and details of the feature words and feature values will be introduced in Section 4.

3.3. System training in the proposed method

The training process is divided into two parts: row units training and MDCA unit training. We must train the row units first and then train the MDCA unit. After we input a training sentence into the system, each row unit of the system will be adjusted based on the target sentiment value T of the sentence and the determinant value $|M|$ of the matrix M. The adjustment rate r will be calculated by

$$r = \frac{T}{\sqrt[n]{|M|}} \tag{7}$$

For real numbers, when the n is an even number, the object of the square root operation must be a nonnegative number. So we must make $|M|$ a positive number. A simple method is as follows: we can add a positive bias b on all the elements of the diagonal of the matrix. b must be large enough to keep the matrix determinant a positive value at all times. The new matrix with bias, represented by $M^{'}$, will be represented by

$$M^{'} = M + b \times E$$
$$= \begin{bmatrix} m_{11}+b & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22}+b & \dots & m_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & m_{ij} & \dots \\ \dots & \dots & \dots & \dots \\ m_{n1} & m_{n2} & \dots & m_{nn}+b \end{bmatrix} \tag{8}$$

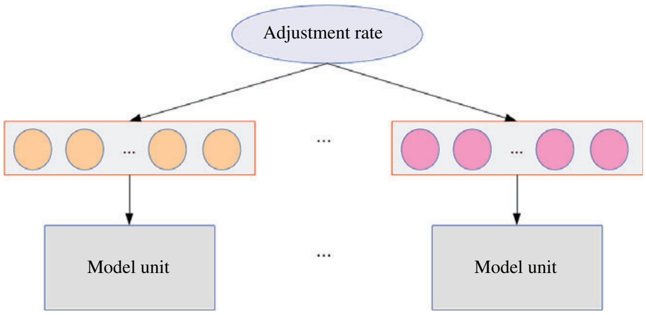


Fig. 3. System training

The rate r and the target sentiment value T should also be changed into (9) and (10):

$$T^{'} \approx T + \boldsymbol{b} \tag{9}$$

$$r^{'} = \frac{T^{'}}{\sqrt[n]{|M^{'}|}} \tag{10}$$

If we want the determinant value to be positive, the matrix must meet the condition that in each row the element value on the diagonal must be larger than the sum of the absolute values of the other elements [52–54]. In the system case, if all the elements in the matrix M do not exceed 1, we can set b with (11):

$$\boldsymbol{b} > n - 2 \tag{11}$$

After getting the adjustment rate r, r will adjust all the elements of the matrix, and then the adjusted element values will be passed to the row units as training targets. The procedure is illustrated in Fig. 3. We can treat this step as a process to train a regression model for matrix M as input and matrix M_t as the output (M_t is shown in (12)). Using the BPNN algorithm [44], the weights in the row units will be changed:

$$M_t = M^{'} \times r^{'} - b \times E \tag{12}$$

In the row unit training, the training rate r is based on $T + \boldsymbol{b}$ and $|M^{'}|$, while in most cases $T^{'}$ does not equal $T + \boldsymbol{b}$ [51]. So after we train the row units, we must train the MDCA unit to do the adjustment. That means, we train a regression to fit $\sqrt[n]{|M^{'}|} - b$ generated by the trained row units and the matrix generation unit to approach the real sentiment value T.

4. Experiment

4.1. Environment setting

We use a method that has been used by the research group of Stanford Treebank Dataset [8] to represent the sentiment value. Using the continuous value from 0 to 1, where 0 means the most negative and 1 means the most positive, the sentiment value of a sentence or a word will be represented by a value such as a probability. Since the sentences in Stanford Treebank Dataset are all short sentences, we also use the Stanford Treebank Dataset as the training corpus. The dataset is divided into three parts as in Ref. 47: of the 11 855 sentences, 8544 sentences are used for training the row units, 1101 sentences are used for training the MDCA unit, and the last 2210 sentences are used for testing. The sentiment value set on each sentence is uniformly distributed.

We use NN to build and train the row units and MDCA unit, and use the PyBrain toolkit to construct and train the NN units [55]. In each NN, the output layer uses the sigmoid function [56,57] layer, and the hidden layer uses a linear function. We train the NN units until convergence but not exceeding the maximum epochs 30. The sentiment matrix is in three dimensions corresponding modifying *subject*, *direct object*, and *indirect object*, and with bias b set to 3

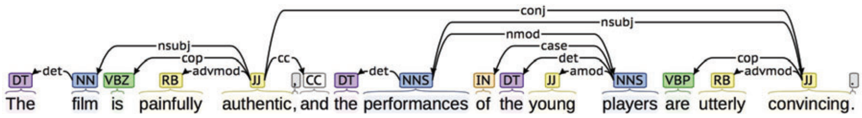


Fig. 4. Dependency parsing example of CoreNLP (this figure was generated by the Stanford CoreNLP online visual toolkit (http://nlp.stanford.edu:8080/corenlp/process))

adding to each element on the diagonal of the matrix. For each row unit, the NN layers and nodes are set by RULayer = [3, 3, 3, 3, 3, true], which means one layer with three feature nodes for input, three hidden layers with three nodes in each hidden layer, and one output layer with three output nodes, and with NNBias = True. The MDCA unit is set by MDCALayer = [1, 3, 3, 3, 1, true]. All the parameters of NNs are initialized at random at the beginning of each training.

We use the dependency parsing module of the Stanford CoreNLP Toolkit [58] to abstract the modifying dependency relations. For example, the dependency parsing result of the sentence 'The film is painfully authentic, and the performances of the young players are utterly convincing' is illustrated in Fig. 4. Each dependency relation is a directive from a source word to a target word: e.g. the relation *advmod* (adverb modifier) from source word 'convincing' to target word 'utterly'. We first get all the vector feature words (VFWs): both source and target words of all the subject-relative relations such like *nsubj* (nominal subject), *nsubjpass* (passive nominal subject), etc. and passed to the *subject* row of the vector. Then we get target words of all the direct-object-relative relations and pass them to the *direct object* row, and get target words of all the indirect-object-relative relations and pass them to the *indirect object* row. Then we abstract all the sentiment-relative and modifying-relative relations such as *advmod* (adverb modifier), *amod* (adjectival modifier), *neg* (negation modifier), etc., and then use the target words as matrix feature words (MFWs). The feature words will be passed to the matrix row, where the vector feature words of the row are the nearest in the dependency parsing tree. After this step, we can get the initial vector feature words and initial MFWs from the example sentence. The feature words are shown in (13) and (14), using '☐' to represent the empty position:

$$\mathrm{VFW} = \begin{bmatrix} \mathrm{vfw}_{\mathrm{subj}} \\ \mathrm{vfw}_{\mathrm{dobj}} \\ \mathrm{vfw}_{\mathrm{iobj}} \end{bmatrix} = \begin{bmatrix} \text{film, performance} \\ \square \\ \square \end{bmatrix} \tag{13}$$

$$\mathrm{MFW} = \begin{bmatrix} \mathrm{mfw}_{\mathrm{subj}} \\ \mathrm{mfw}_{\mathrm{dobj}} \\ \mathrm{mfw}_{\mathrm{iobj}} \end{bmatrix} = \begin{bmatrix} \text{painfully, authentic, young, utterly, convincing} \\ \square \\ \square \end{bmatrix} \tag{14}$$

Based on the MFWs, we can get the feature values F_i on each row, see (15): the average sentiment values of the matrix feature words on each row AV $G(\mathrm{mfw}_i)$; the average sentiment values of the negative modifying words on each row AV $G(\mathrm{neg}_i)$; and the average of sentiment values of all the words in the sentence AV $G(S)$. The sentiment value of each word and the full sentence can be searched from the sentiment dictionary of the Stanford Treebank Dataset [8]. Empty positions of the matrix represented by '☐' will be set by 0.5, which means neutral or no sentiment modification. These matrix feature values will be used as the input of each row unit, and the sentiment value of the full sentence will be used as the target value T:

$$F_i = [\mathrm{AV}\ \ G(\mathrm{mfw}_i), \mathrm{AV}\ \ G(\mathrm{neg}_i), \mathrm{AV}\ \ G(S)], \quad (i = \mathrm{subj}, \mathrm{dobj}, \mathrm{iobj}) \tag{15}$$

The experiment is divided into three parts: regression, three-class classification, and five-class classification. The regression evaluation is based on mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum_{1}^{n} \mid e_i \mid = \frac{1}{n}\sum_{1}^{n} \mid f_i - y_i \mid \tag{16}$$

Here, e_i means the error between the system output value f_i and the target value y_i. In our experiment, the MAE value is not allowed to exceed 0.2, which means that the sentiment value error will not exceed a class distance of five classes of the Stanford Treebank: 'very negative', 'negative', 'neutral', 'positive', and 'very positive'.

The classification evaluation is compared with the simplified SCLM and the sentiment computing module of Stanford CoreNLP with a trained model on the official net. We compare the performance of three-class and five-class classifications among the three methods. The three classes include 'negative', 'neutral', 'positive', with the sentiment value range [0, 0.4), [0.4, 0.6), and [0.6, 1]. The five classes include 'very negative', 'negative', 'neutral', 'positive', and 'very positive', with the sentiment value range [0, 0.2), [0.2, 0.4], [0.4, 0.6], [0.6, 0.8], and [0.8, 1]. The evaluation of classification is based on precision, recall, $F1$ value, macro-precision, macro-recall and macro-$F1$ value, given by (17)–(22) respectively:

$$\mathrm{Precision}_c = \frac{T_c}{T_c + F_c} \tag{17}$$

$$\mathrm{Recall}_c = \frac{T_c}{\mid \mathrm{Set}_c \mid} \tag{18}$$

$$F1_c = \frac{2\mathrm{Precision}_c \times \mathrm{Recall}_c}{\mathrm{Precision}_c + \mathrm{Recall}_c} \tag{19}$$

$$\text{Macro-precision} = \frac{1}{\mid C \mid}\sum_{c \in C} \mathrm{Precision}_c \tag{20}$$

$$\text{Macro-recall} = \frac{1}{\mid C \mid}\sum_{c \in C} \mathrm{Recall}_c \tag{21}$$

$$\text{Macro-}F1 = \frac{2 \times \text{Macro-precision} \times \text{Macro-recall}}{\text{Macro-precision} + \text{Macro-recall}} \tag{22}$$

In these formulas, c is one of the sentiment classes and C is the universal set of all the sentiment classes. $|C|$ is the number of all the classes (e.g. in the five-class experiment $|C| = 5$), and $|\mathrm{Set}_c|$

Table I. Three-class experimental results

	SCLM	SCLM(b)	CoreNLP	Best 3c Model
$\text{Precision}_{\text{positive}}$	0.7273	0.8095	0.7761	**0.8494**
$\text{Precision}_{\text{neutral}}$	0.1772	0.1772	0.3881	0.1914
$\text{Precision}_{\text{negative}}$	0.6000	0.6000	0.6612	**0.8452**
Macro-precision	0.5015	0.5289	0.6085	**0.6287**
$\text{Recall}_{\text{positive}}$	0.0088	0.0187	0.7701	0.1551
$\text{Recall}_{\text{neutral}}$	0.9974	0.9949	0.2005	0.9229
$\text{Recall}_{\text{negative}}$	0.0066	0.0033	0.8026	0.1557
Macro-recall	0.3376	0.3389	0.5911	0.4112
$F1_{\text{positive}}$	0.0174	0.0366	0.7731	0.2623
$F1_{\text{neutral}}$	0.3010	0.3008	0.2644	0.3170
$F1_{\text{negative}}$	0.0130	0.0065	0.7251	0.2630
Macro-$F1$	0.4035	0.4131	0.5996	0.4972

Table II. Five-class experimental results

	CoreNLP	Best 5c Model
$\text{Precision}_{\text{verypos}}$	0.6707	**0.8462**
$\text{Precision}_{\text{positive}}$	0.4095	0.3290
$\text{Precision}_{\text{neutral}}$	0.3881	0.2078
$\text{Precision}_{\text{negative}}$	0.4476	**0.4509**
$\text{Precision}_{\text{veryneg}}$	0.4947	**0.6538**
Macro-precision	0.4821	**0.4975**
$\text{Recall}_{\text{verypos}}$	0.2807	0.0276
$\text{Recall}_{\text{positive}}$	0.5901	0.2961
$\text{Recall}_{\text{neutral}}$	0.2005	0.6992
$\text{Recall}_{\text{negative}}$	0.7156	0.2686
$\text{Recall}_{\text{veryneg}}$	0.1685	0.1219
Macro-recall	0.3911	0.2827
$F1_{\text{verypos}}$	0.3958	0.0534
$F1_{\text{positive}}$	0.4835	0.3117
$F1_{\text{neutral}}$	0.2644	0.3204
$F1_{\text{negative}}$	0.5508	0.3366
$F1_{\text{veryneg}}$	0.2513	0.2054
Macro-$F1$	0.4319	0.3605

is the number of sentences belonging to the c sentiment class (e.g. in the five-class experiment, when c is 'very positive', $|\text{Set}_c| = 399$, which means that there are 399 sentences belonging to the 'very positive' class). T_c is the number of the elements that are correctly classified into the class of c (True c label), while F_c are the element counts that are classified into c but in fact do not belong to c (False c label).

4.2. Results and analysis We trained 31 models, with all the parameters initialized randomly at the beginning of each training. Each training and test will spend about 5 h. Then we choose two models: one has the best performance in the three-class classification (represented as 'Best 3c Model') and the other has the best performance in the five-class classification (represented as 'Best 5c Model'). The three-class experimental results are listed in Table I and the five-class experimental results are listed in Table II. We also list the three-class classification test results of the best five-class model in Table III. The SCLM and the Best 3c model have no successful results in the five-class experiment because the precision and recall values of some subclasses are 0. So we only compare the five-class experimental results only between CoreNLP and the proposed method. The MAE value of the best three-class model is 0.1984, and the MAE value of the best five-class model is 0.1787.

The classification results show that the proposed method can deal with the five-class classification task successfully, and on most evaluation parameters of three classes, the proposed method has improved in most of the performances than the two methods

Table III. Best 5c Model on three-class classification test

	Precision	Recall	$F1$
Positive	**0.7966**	0.4136	0.5445
Neutral	0.2078	0.6992	0.3204
Negative	**0.7506**	0.3531	0.4802
Macro-	0.5850	0.4886	0.5325

Table IV. Performance improvements on the three-class test

	Best 3c Model		Best 5c Model	
	SCLM	SCLM(b)	SCLM	SCLM(b)
$\text{Precision}_{\text{positive}}$	0.1221	0.0399	0.0693	−0.0129
$\text{Precision}_{\text{neutral}}$	**0.0142**	**0.0142**	0.0306	0.0306
$\text{Precision}_{\text{negative}}$	**0.2452**	**0.2452**	0.1506	0.1506
Macro-precision	0.1272	0.0998	0.0835	0.0561
$\text{Recall}_{\text{positive}}$	0.1463	0.1364	0.4048	0.3949
$\text{Recall}_{\text{neutral}}$	−0.0745	−0.0720	−0.2982	−0.2957
$\text{Recall}_{\text{negative}}$	0.1491	0.1524	0.3465	0.3498
Macro-recall	0.0736	0.0723	0.1510	0.1497
$F1_{\text{positive}}$	0.2449	0.2257	0.5271	0.5079
$F1_{\text{neutral}}$	**0.0160**	**0.0162**	0.0194	0.0196
$F1_{\text{negative}}$	**0.2500**	**0.2565**	0.4672	0.4737
Macro-$F1$	0.0937	0.0841	0.1290	0.1194

based on the simplified SCLM (only the recall of the neutral in the three-class model is not improved). The improvements are shown in Table IV. The precision of Best 3c model has been improved from 1.42% to 24.52%, the $F1$ value has been improved from 1.60% to 25.65%. When compared with the Stanford CoreNLP in both three classes and five classes, there are also some cases in which the new method shows better performance (these cases have been marked in bold). The MAE values of all these models did not exceed 0.2.

The results show that the proposed method is good at classification and attaches more importance to precision. The results of the five-class experiment also showed that the proposed method is very good at dealing with extreme sentiment ('very positive' and 'very negative'). Compared with CoreNLP in the five-class experiment, the precision of 'very positive' has been improved by 17.55% and the precision of 'very negative' has been improved by 15.91%. However, the recall and $F1$-value of CoreNLP are much better than those of the proposed method. The recall performance of the proposed method still needs improvement.

4.3. Discussion The recall values of the nonneutral classes of the proposed method are lower than those of CoreNLP. One of the reasons is that the proposed method depends on selected sentiment features and sentiment value of the feature words. In the experiment, we chose three common features ('subject', 'direct object' and 'indirect object') to construct the model. If the sentiment distribution of a sentence does not cover most of these feature parts, or if the proportion of sentiment distribution of the feature words is much less than the others, the model will make mistakes. Another reason is that, in the experiment, the empty positions of the matrix represented by '□' are all set to 0.5; if the selected features cannot represent most of the sentiment distribution of the sentence, the measurement result will tend to the neutral class. This will cause lower recalls of the nonneutral classes and higher recalls of the neutral classes. And also, the randomly generated initial values of the NNs also cause uncertainties in the model's performance.

Because of these reasons, we plan to fix these defects from three perspectives. First, we will try more feature designs by adding or changing features. Second, we will try to use some complex method for the value-filling of the initial matrix, e.g. try to integrate information of both feature elements and nonfeature elements. Third, based on the current trained models, we will try to determine the relationship between the initial values of the NNs and the trained model performance, and then optimize the initialized setting.

5. Conclusion and Future Work

In this paper, an SA method was proposed to deal with sentiment measurement and classification using a modifying-matrix-based language model. The regression result shows that the deviation between the output sentiment and the target sentiment does not exceed a class distance of five sentiment class range. Classification experiment showed that the proposed method has improved most performances compared to the simplified SCLM, and in some cases it has a higher precision performance. However, the recall performance of the proposed method still needs improvement.

The advantage of the SCLM and the proposed method is that they treat all the words that contain sentiment information as modifying, including negative-modifying. So we do not need to parse the complex syntax rules of negative sentences. However, there is a disadvantage in that the proposed method ignores the negation of the same vector feature position. For example, in the setting of the experiment, the difference between ‘I don’t really like it’ and ‘I really don’t like it’ cannot be recognized, because the word ‘really’ and ‘do not’ are in the same modifying position.

Because of these we will focus on improving the recall performance and try to recognize the sentiment difference in finer granularity in future work. We will focus on the relationship between the initial values of the system and the performance after training. And also, we will try to propose a complex operational model for the matrices of SCLM to take advantage of the simplified SCLM.

Acknowledgment

This work was partially supported by JSPS KAKENHI, grant number 15H01712.

References

(1) Serrano-Guerrero J, Olivas JA, Romero FP, Herrera-Viedma E. Sentiment analysis: a review and comparative analysis of web services. *Information Sciences* 2015; **311**:18–38.

(2) Ren F, Wu Y. Predicting user-topic opinions in twitter with social and topical context. *IEEE Transactions on Affective Computing* 2013; **4**(4):412–424.

(3) Vinodhini G, Chandrasekaran R. Sentiment analysis and opinion mining: a survey. *International Journal of Advanced Research in Computer Science and Software Engineering* 2012; **2**(6):282–292.

(4) Leite I, Martinho C, Paiva A. Social robots for long-term interaction: a survey. *International Journal of Social Robotics* 2013; **5**(2):291–308.

(5) Breazeal C. Affective Interaction between Humans and Robots. In *Advances in Artificial Life*. Kelemen J, Sosík P (eds). Springer: Berlin Heidelberg; 2001; 582–591.

(6) Chandrakala S, Sindhu C. Opinion Mining and sentiment classification: a survey. *ICTACT Journal on Soft Computing* 2012; **3**(1):420–427.

(7) Han Z, Ren F, Miao D. Synthetic and computational language model for interactive dialogue system. *European Conference on Data Mining 2015 and International Conferences on Intelligent Systems and Agents 2015 and Theory and Practice in Modern Computing 2015,* New York, NY, 2015; 73–80.

(8) Socher R, Bauer J, Manning CD, Ng AY. Parsing with compositional vector grammars. *Proceedings of the ACL Conference*, Citeseer, 2013.

(9) Stavrianou A, Andritsos P, Nicoloyannis N. Overview and semantic issues of text mining. *ACM SIGMOD Record* 2007; **36**(3):23–34.

(10) Toutanova K, Klein D, Manning CD, Singer Y. Feature-rich part-of-speech tagging with a cyclic dependency network. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology,* Vol. 1, 2003; 173–180. Association for Computational Linguistics.

(11) De Marneffe M-C, MacCartney B, Manning CD. Generating typed dependency parses from phrase structure parses. *Proceedings of the LREC*, Vol. 6, 2006; 449–454.

(12) Nakagawa T, Inui K, Kurohashi S. Dependency tree-based sentiment classification using crfs with hidden variables. *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, June 2–4, 2010, Los Angeles, California*, 786–794.

(13) Liu Z, Yu W, Chen W, Wang S, Wu F. Short text feature selection for micro-blog mining. *2010 International Conference on Computational Intelligence and Software Engineering* (*CiSE*), 2010; 1–4. IEEE.

(14) Song Y, Wang H, Wang Z, Li H, Chen W. Short text conceptualization using a probabilistic knowledgebase. *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence,* Vol. 3, 2011; 2330–2336. AAAI Press.

(15) Dong Z, Dong Q. HowNet-a hybrid language and knowledge resource. *Proceedings of the 2003 International Conference on Natural Language Processing and Knowledge Engineering,* 2003; 820–824. IEEE.

(16) Miller GA. WordNet: a lexical database for English. *Communications of the ACM* 1995; **38**(11):39–41.

(17) Pak A, Paroubek P. Twitter as a corpus for sentiment analysis and opinion mining. *LREC* 2010; **10**:1320–1326.

(18) Quan C, Ren F. Feature-level sentiment analysis by using comparative domain corpora. *Enterprise Information Systems* 2014; **10**(5):505–522.

(19) Wenyin L, Quan X, Feng M, Qiu B. A short text modeling method combining semantic and statistical information. *Information Sciences* 2010; **180**(20):4031–4041.

(20) Miao D, Zheng F, Zhang Z, Gao C. News topic detection approach on Chinese microblog. *Computer Science* 2012; **1**:033.

(21) Wang L, Miao D, Zhao C. Chinese emotion recognition based on three-way decisions. In *Rough Sets and Knowledge Technology*. Ciucci D, Wang G, Mitra S, Wu WZ (eds). Springer: Cham; 2015; 299–308.

(22) Davidov D, Tsur O, Rappoport A. Enhanced sentiment learning using twitter hashtags and smileys. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 2010; 241–249. Association for Computational Linguistics.

(23) Thelwall M, Buckley K, Paltoglou G, Cai D, Kappas A. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology* 2010; **61**(12):2544–2558.

(24) Cheng X, Yan X, Lan Y, Guo J. BTM: topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering* 2014; **26**(12):2928–2941.

(25) Jin O, Liu NN, Zhao K, Yu Y, Yang Q. Transferring topical knowledge from auxiliary long texts for short text clustering. *Proceedings of the 20th ACM international Conference on Information and Knowledge Management*, 2011; 775–784. ACM.

(26) O’Connor B, Balasubramanyan R, Routledge BR, Smith NA. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM* 2010; **11**(122–129):1–2.

(27) Pröllochs N, Feuerriegel S, Neumann D. Negation scope detection in sentiment analysis: Decision support for news-driven trading. *Decision Support Systems* 2016; **88**:67–75.

(28) Pröllochs N, Feuerriegel S, Neumann D. Detecting negation scopes for financial news sentiment using reinforcement learning. *Hawaii International Conference on System Sciences*, 2016; 1164–1173.

(29) Huang PC, Wu JS, Lee CN. Negative emotion event detection for Chinese posts on facebook. *International Conference on Cloud Computing and Big Data*, 2015; 329–335.

(30) Zhifei Z, Duoqian M, Jianyun N, Xiaodong Y. Sentiment uncertainty measure and classification for negative sentences. *Journal of Computer Research and Development* 2015; **52**(8):1806–1816.

(31) Huang WY, Pao TL. A study on the combination of emotion keywords to improve the negative emotion recognition accuracy. *6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012)*, Taipei, 2012; 499–503.

(32) Zhang X, Wang H, Li L, Zhao M, Li Q. Negative emotion recognition in spoken dialogs. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Sun M, Liu Z, Zhang M, Liu Y (eds). Lecture Notes in Computer Science, vol 9427. Springer: Cham; 2015; 103–115.

(33) Angeli G, Manning CD. NaturalLI: natural logic inference for common sense reasoning. *Proceedings of the Empirical Methods in Natural Language Processing* (*EMNLP*), 2014; 534-545.

(34) Socher R, Huval B, Manning CD, Ng AY. Semantic compositionality through recursive matrix-vector spaces. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012; 1201–1211. Association for Computational Linguistics.

(35) Bengio Y, Ducharme R, Vincent P, Janvin C. A neural probabilistic language model. *Journal of Machine Learning Research* 2003; **3**:1137–1155.

(36) Mnih A, Teh YW. A fast and simple algorithm for training neural probabilistic language models, 2012. arXiv preprint arXiv:1206.6426.

(37) Kotsiantis S. Supervised machine learning: a review of classification techniques. *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, 2007; 3–24. IOS Press.

(38) Vaggi M, Prandin S, Bellentani G, Costantini A, Labib A. Supervised Neural Network Algorithms: a proved successful methodology to identify unconventional layers opens new potential development scenarios of Abu Qir fields in the Mediterranean Sea offshore Egypt. *Offshore Mediterranean Conference and Exhibition*, 2013.

(39) Dayhoff JE. *Neural Network Architectures: An Introduction*. Van Nostrand Reinhold Co.: New York; 1990.

(40) Anderson JA. *Neurocomputing*, vol. **2**. MIT Press: Cambridge, MA; 1993.

(41) Bengio Y. Learning deep architectures for AI. *Foundations and Trends® in Machine Learning* 2009; **2**(1):1–127.

(42) Ngiam J, Khosla A, Kim M, Nam J, Lee H, Ng AY. Multimodal deep learning. *Proceedings of the 28th International Conference on Machine Learning* (*ICML-11*), 2011; 689–696.

(43) Schmidhuber J. Deep learning in neural networks: an overview. *Neural Networks* 2015; **61**:85–117.

(44) Hecht-Nielsen R. Theory of the backpropagation neural network. *International Joint Conference on Neural Networks* (*IJCNN*), 1989; 593–605. IEEE.

(45) Ham FM, Kostanic I. *Principles of Neurocomputing for Science and Engineering*. McGraw-Hill Higher Education: New York; 2000.

(46) Socher R, Pennington J, Huang EH, Ng AY, Manning CD. Semi-supervised recursive autoencoders for predicting sentiment distributions. *Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27–31 July 2011, John Mcintyre Conference Centre, Edinburgh, UK, A Meeting of Sigdat, A Special Interest Group of the ACL*, 2011; 151–161.

(47) Socher R, Perelygin A, Wu JY, Chuang J, Manning CD, Ng AY, Potts C. Recursive deep models for semantic compositionality over a Sentiment Treebank. *2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics (ACL)*, 2013; 1631–1642.

(48) Yasuda M, Tanaka K. Approximate learning algorithm for restricted Boltzmann machines. *2008 International Conference on Computational Intelligence for Modelling Control & Automation,* 2008; 692–697. IEEE.

(49) Paola J, Schowengerdt R. A review and analysis of backpropagation neural networks for classification of remotely-sensed multispectral imagery. *International Journal of Remote Sensing* 1995; **16**(16):3033–3058.

(50) Wilkinson JH, Bauer FL, Reinsch C. *Linear Algebra*, vol. **2**. Springer: Berlin Heidelberg; 2013.

(51) Golub GH, Van Loan CF. *Matrix Computations*, vol. **3**. JHU Press: Baltimore, MD; 2012.

(52) Jin S, Xiong F, Li Q. Conditions and judgements of matrices with positive real eigenvalues. *Journal of Chongqing University of Technology (Natural Science)* 2011; **25**:117–119.

(53) Meyer CD. *Matrix Analysis and Applied Linear Algebra*, vol. **2**. Siam: Philadelphia, PA; 2000.

(54) Bhatia R. *Positive Definite Matrices*. Princeton University Press: Princeton, NJ; 2009.

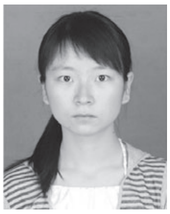
(55) Schaul T, Bayer J, Wierstra D, Sun Y, Felder M, Sehnke F, Rückstieß T, Schmidhuber J. PyBrain. *Journal of Machine Learning Research* 2010; **11**:743–746.

(56) Uchimurai S, Harnamotoi Y, Tomitai S. On the effect of the nonlinearity of the sigmoid function in artificial neural network classifiers. *Proceedings of the 1995 IEEE International Conference on Neural Networks, University of Western Australia, Perth, Western Australia, 27 November–1 December 1995*, Vol. 1, 1995; 281. IEEE.

(57) Curry B, Morgan PH. Neural networks, linear functions and neglected non-linearity. *Computational Management Science* 2003; **1**(1):15–29.

(58) Manning CD, Surdeanu M, Bauer J, Finkel J, Bethard SJ, McClosky D. The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014; 55–60.

Zhao Han (Non-member) received the Bachelor's degree from Tongji University (TU), China, in 2012. She further pursued the Master's and doctoral degree courses at TU. She is currently pursuing a double-degree program developed by both TU and Tokushima University. Her research interests include affective computing, computational linguistics, and dialogue system.



Fuji Ren (Member) received the Ph.D. degree in 1991 from the Faculty of Engineering, Hokkaido University. From 1991 he worked with CSK, Japan, where he was a Chief Researcher of NLP. From 1994 to 2000, he was an Associate Professor with the Faculty of Information Science, Hiroshima City University. He became a Professor with the Faculty of Engineering, Tokushima University, in 2001. He has been the President of the AIA International Advanced Information Institute since 2003. He is an Academician of the Engineering Academy of Japan. His research interests include artificial intelligence, language understanding and communication, and affective computing. Prof. Ren is a member of the IEICE, CAAI, IPSJ, JASI, and AAMT, a senior member of the IEEE, a Vice-President of CAAI, and a Fellow of the Japan Federation of Engineering Societies.

Duoqian Miao (Non-member) received the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, in 1997. His research interests include rough sets, granular computing, principal curve, web intelligence, and data mining. Prof. Mio is a committee member of International Rough Sets Society (IRSS), a distinguished member of the China Computer Federation (CCF), a member of the CCF Artificial Intelligence and Pattern Recognition Technical Committee, a committee member of Chinese Association for Artificial Intelligence (CAAI), Chair of CAAI Rough Sets and Soft Computing Technical Committee and standing member of CCAI Machine Learning Technical Committee, and Chair of Shanghai Computer Society (SCA) Computing Theory and Artificial Intelligence Technical Committee.