Soft Comput (2018) 22:4241–4249
https://doi.org/10.1007/s00500-017-2772-7

FOCUS

# Automatic scoring system for short descriptive answer written in Korean using lexico-semantic pattern

**Jeong-Eun Kim[1] · Kinam Park[1] · Jeong-Min Chae[1] · Hong-Jun Jang[1] · Byoung-Wook Kim[1,2] · Soon-Young Jung[1]**

Published online: 17 August 2017


**Abstract** The researches on the automatic scoring system for English descriptive answers have been actively performed, but there are not so many researches on the automatic scoring system for Korean descriptive answers. In this paper, we propose an scoring method based on lexico-semantic pattern (LSP), which is known to be a good solution for the morphologically rich Korean language. In the proposed method, postposition information is utilized as an important tool for finding the meaning differences in Korean. In addition to using LSP, we also applied a synonym dictionary as a meaning extension approach to improve recall performance in scoring student's answer. Our experimental result shows that the proposed system performs better than the existing noun-keyword-based system by 0.137. Also, the best performance could be obtained by using a synonym dictionary.

Communicated by J. Park.

✉ Soon-Young Jung
jsy@korea.ac.kr

Jeong-Eun Kim
marite76@korea.ac.kr

Kinam Park
spknn@korea.ac.kr

Jeong-Min Chae
onacloud@korea.ac.kr

Hong-Jun Jang
hongjunjang@korea.ac.kr

Byoung-Wook Kim
byoungwook.kim@inc.korea.ac.kr

[1] Department of Computer Science and Engineering, Korea University, Seoul, Korea

[2] Creative Informatics & Computing Institute, Korea University, Seoul, Korea

**Keywords** Lexical semantic pattern · Synonym · Korean short descriptive answer · Automatic scoring

## 1 Introduction

In school, most of tests for evaluating the achievement of learning have been performed using simple type questions such as true/or false question, multiple-choice question or simple word question. However, in spite of its convenience, the evaluation using simple type question is known as being inappropriate in assessing the achievement of learning about the process knowledge, complex knowledge, logical thinking ability, etc.

It is addressed generally that descriptive evaluation methods are appropriate in learning evaluation about those knowledge. However, scoring manually the descriptive answers is not only time-consuming and labor-consuming, but also has a difficulty in maintaining the consistency of scoring, even though they use a grading rubrics. In order to solve these problems, various automatic scoring techniques for the descriptive assessment have been studied by this time. The typical approach of researches on automatic scoring of English descriptive answer is to extract meaningful words from the descriptive answers and predict scores based on the extracted words.

Representative methods to extract meaningful words from text are a conceptual indexing method that reflects the semantic similarity of words to index information and a method of predicting the semantic similarity by reconstructing vector space from whole semantic coordinate system to statistically meaningful coordinate system using latent semantic analysis (LSA) (Dobrov et al. 1997; Loukachevitch and Dobrov 2002; Valenti et al. 2003).

The similarity measurement method based on maximum entropy or regression method based on the support vector

machine (SVM) was proposed and produced meaningful results as a method of predicting the score based on extracted words (Chakrabartty and Cauwenberghs 2007). As such, in scoring automatically English descriptive answers, various researches that utilize the meaning of words have been performed.

In the study on the automatic scoring of the Korean descriptive answer, similar methods as the English methods have been suggested, but the results have not been satisfactory yet. The reason is that it is difficult to apply the conceptual index method, which is used in automatic scoring of English descriptive answers, to Korean because the method of determining the meaning of words in English sentences is different from Korean.

More specifically, the English is a structured language and its meaning is easily determined according to the position of a word such as a verb and a noun, but in the case of Korean, the order of the words and the propositional particles that combines them play a role in determining its meaning. Furthermore, the frequency of the propositional particles in a sentence is high, and variations in combination with other words are also diverse. Due to this characteristic of Korean, it is difficult to expect satisfactory performance to apply the automatic evaluation technique of English narrative answer to Korean.

In this paper, we propose a method of automatic scoring of the short descriptive answer considering these characteristics of Korean. The proposed method uses the LSP that is designed in this study as a method to express the Korean features that affect the accuracy of the scoring and the expanded concepts of the words contained in the sentence.

## 2 Related works

### 2.1 Automatic scoring system for English short descriptive answer

The various researches on the automatic scoring system for English descriptive answers have been actively performed. Burrows et al. (2015) proposed a method of extracting meaningful words from answer paper and calculating its score based on the extracted words. Jadidinejad and Mahmoudi (2014) presented a method to assess based on the presence or absence of each concept made through the concept mapping of assessment papers. Burstein and Chodorow (1999) proposed a scoring method using the lexical conceptual structure representation, which consists of the concept-based lexicon and the concept grammar extracted from the training set. The Automatic Text Marker (ATM) proposed by Burrows et al. (2015) and Callear et al. (2001) divides the answers of teachers and students into a minimal concept list, counts the number of concepts and scores them. The C-rater proposed by Clariana and Wallace (2007) is a scoring system that calculates similarity of sentence-level concepts between student's answer and teacher's answer and then assigns scores according to its similarity.

### 2.2 Automatic scoring system for Korean short descriptive answer

The researches on the automatic scoring system for English descriptive answers have been actively performed, but there are not so many researches on the automatic scoring system for Korean descriptive answers. Jang et al. (2014) explored a Korean automatic scoring system for short- and free-text responses. The system builds a token-based scoring template and grades the descriptive answers based on the template for increasing the coverage of the automatic scoring process. Cho et al. (2005) proposed an intelligent scoring system based on the semantic kernel and the Korean WordNet. This system tried to increase the accuracy by expanding vocabularies (thesaurus and synonyms) appeared in the student's answers and in the correct answers simultaneously. However, this system did not treat the semantic correlation in accordance with the position of the syntactic words. Bae (2013) proposed a method of automatic scoring of Korean descriptive answers using the predicate normalization. The method was focused on improving the accuracy of the automatic scoring by using the predicate normalization method that converts verbs and adjectives into noun forms according to a set of rules. The accuracy of discrimination of correct answer was increased due to improvement on the accuracy of calculation of similarity, but it did not give a great influence on the discrimination of incorrect answers.

Kim et al. (2014) proposed an automatic scoring method for descriptive answers based on the LSP. However, the method did not show satisfactory performance by using only the nouns extracted from the correct answer without applying the concept extension technique. And the function of giving a partial score was not included, and practical use is difficult in school.

### 2.3 Lexico-semantic pattern

As information on the web has grown dramatically, many researches on the efficient web search technique have been performed. Among such researches, there have been attempts to improve web search performance using various semantic-based information. The lexico-semantic pattern (LSP) was proposed as a method to represent and structure the semantic information needed in figuring out user's intent represented in text (Jacobs et al. 1991). LSP treats the correlation between the meaning of the lexical units and the structure of the language or syntax. LSP can be used to normalize the various query expressions with the same intent into smaller groups through the use of the semantic pattern (Cruse 1986). And

thereby, it can reduce the amount of data processing by reducing the number of queries considered.

As mentioned earlier, the LSP has been used to identify and express as clearly as possible the user's intent expressed in text in the field of web search and Q&A systems.

In this study, we use the LSP theory to identify and express clearly the correct descriptive answer presented by the teacher. Expressing the correct answer as an LSP makes it possible to grasp the meaning of the answer more precisely by expressing the keywords of the answer itself as well as expanding the concept of keywords. The resulting LSP will be used to increase the accuracy of the grading of the student's answer.

## 3 Automatic short answer scoring method

The proposed automatic scoring method for the short descriptive answer written in Korean is shown in Fig. 1. The method is composed of two parts. One is the part for generating the LSPs from the teacher's answers and another is the part for scoring the student's answer using the LSPs of the teacher's answers.

The role of each module is as follows: (i) PaMB-list generation module converts the teacher's answer or the student's answers into PaMs in 'Morpheme/POS' format. And then, it generates its bigram (PaMB) list using PaMs (see Sect. 3.1). (ii) The PaMB weighting module generates a weighted PaMB (w-PaMB) list by weighting the PaMBs in PaMB-list based on the appearance frequency of PaMB (see Sect. 3.2). (iii) The Transforming into LSPs module converts w-PaMB-lists into LSPs (see Sect. 3.3). (iv) The Scoring module scores the student's answer represented as the student's PaMB-list using the LSPs of the teacher's answers (see Sect. 3.4).

### 3.1 PaMB-list generation

In this subsection, we describe the process to generate the POS-annotated Morpheme Bigram (PaMB) lists from the teacher's correct answers.

In the first step for generating the PaMB-list, the morphological analysis on teacher's answer is performed. The morphological analyzer used in this study is ETRI's ESTkNLP (2013).

An example of the results of morphological analysis of the teacher correct answers is shown in Fig. 2. The correct answer, '*keullaieonteuga yocheonghamyeon seobeoneun eungdabhanda* (The server responds when the client makes a request)', given by the teacher is converted to '*keullaieonteu/ncn*, *ga/jcs*, *yocheong/ncn*, *ha/xsv*, *myeon/ec*, *seobeo/ncn*, *neun/jx eungdab/ncn*, *ha/xsv*, *nda*/ef' through morphological analysis. Then, the correct answers of the teacher are divided into morphemes (*keullaieonteu*, *ga*, *yocheong*, *ha*, *myeon*, *seobeo*, *neun*, *eungdab*, *ha*, *nda*), and POSs (*ncn*, *jcc*, *xsv*, *ec*, *jx ef, and etc*) are annotated after each morpheme. We call this POS-annotated Morpheme and define as follows:

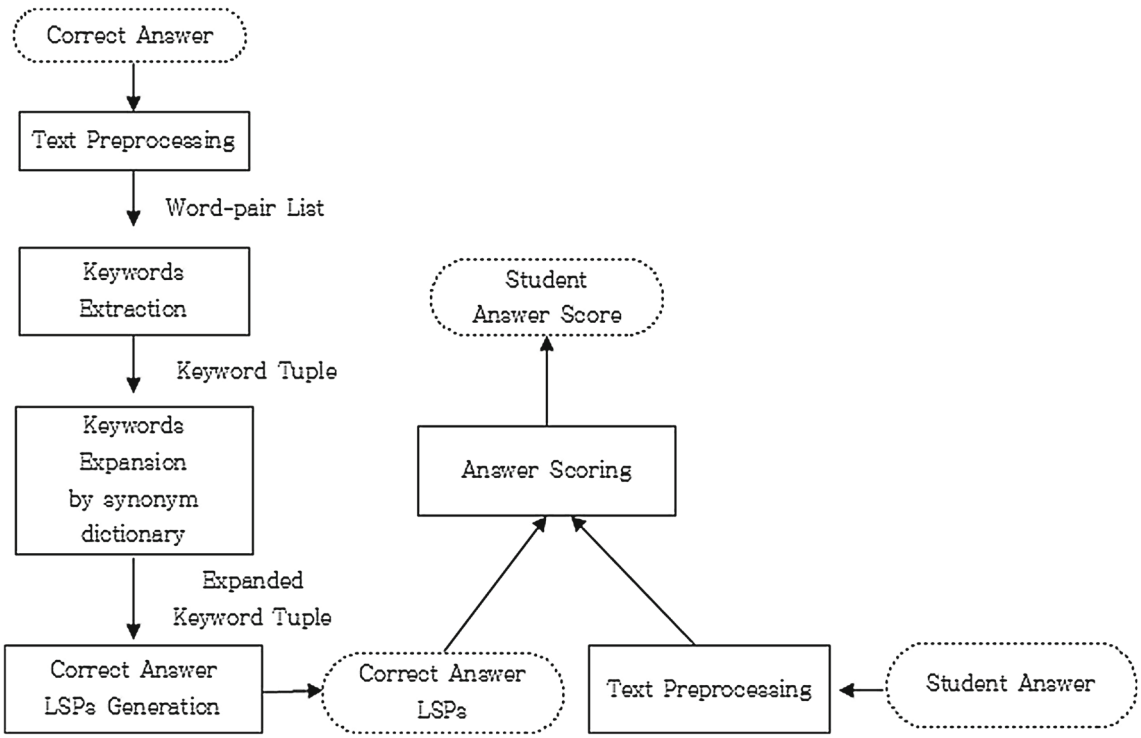


**Fig. 1** The automatic short descriptive answer scoring method

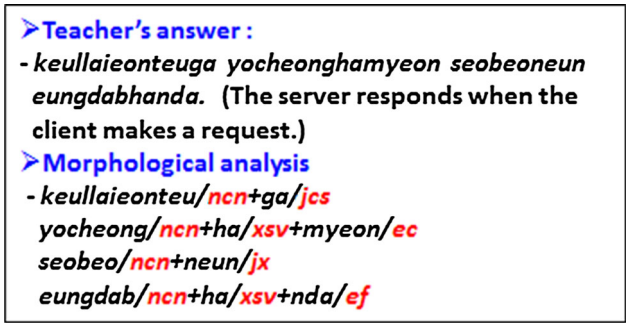


**Fig. 2** An example of morphological analysis

**Definition 1** *POS-annotated Morpheme* (PaM). A PaM is a unit of morpheme and its parts of speech, represented as 'Morpheme/POS'. Given a $\text{PaM}_k$ = 'Morpheme/POS', $\text{PaM}_k$.morp and $\text{PaM}_k$.pos refer to Morpheme and POS, respectively.

A $\text{PaM}_1$ is represented as '*keullaieonteu/ncn*' in which one morpheme '*keullaieonteu*' and its POS '*ncn*' are combined with a '/' symbol in Fig. 3. A PaM is the most basic unit used for transforming the teacher's answer into PaMB-list and is used as the component of PaM bigram which is defined in the following.

**Definition 2** *PaM Bigram* (PaMB). A PaMB is a contiguous sequence of two PaMs from a given sequence of PaMs. Given contiguous two items $\text{PaM}_i$ and $\text{PaM}_j$ $(i < j(=i+1))$, a $\text{PaMB}_i$ is represented as $(\text{PaM}_i{*}\text{PaM}_j)$.

For example, suppose there is one teacher's answer $Ans_1$ = $[\text{PaM}_1, \text{PaM}_2, \text{PaM}_3, \text{PaM}_4]$. PaMBs generated from $Ans_1$ are $\text{PaMB}_1 = (\text{PaM}_1{*}\text{PaM}_2)$, $\text{PaMB}_2 = (\text{PaM}_2{*}\text{PaM}_3)$ and $\text{PaMB}_3 = (\text{PaM}_3{*}\text{PaM}_4)$. More specifically, as shown in Fig. 3, a $\text{PaMB}_1$ is (*keullaieonteu/ncn* **ga/jcs*), which is a form in which $\text{PaM}_1$ '*keullaieonteu/ncn*' and $\text{PaM}_2$ '*ga/jcs*' are connected by '*' symbol. We name $\text{PaM}_1$ as PaMB.front and $\text{PaM}_2$ as PaMB.back.

In this paper, we consider only bigram among n-grams because bigram is the minimum $n$-gram to be able to express the relationship between substantial morpheme and its dependent morpheme (such as the postposition and the suffix) in Korean. The bigram allows to analyze semantic similarity and association more accurately in a single pattern comparison operation.

In the proposed method, the correct answer of the teacher is converted into the PaMB-list as first action. The definition of the PaMB-list is given in Definition 3.

**Definition 3** *PaMB-List* (PaMB-list). Given PaMBs, PaMB-list is an ordered sequence of PaMBs.

Figure 4 shows an example of a PaMB-list. If a given a set $\text{PaMBs} = \{\text{PaMB}_1, \text{PaMB}_2, \ldots, \text{PaMB}_9\}$ the PaMB-list of PaMBs becomes $[\text{PaMB}_1, \text{PaMB}_2, \ldots, \text{PaMB}_9]$. Note that

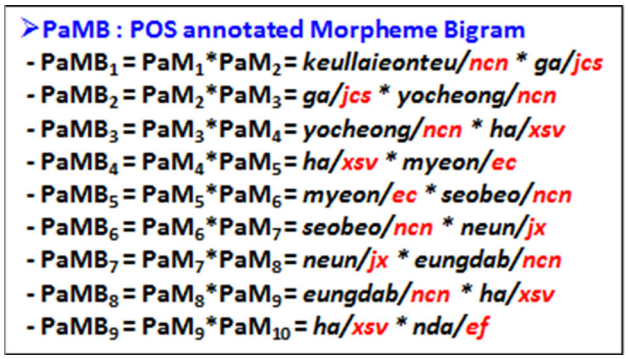


**Fig. 3** Examples of a PaMs and PaMBs

one PaMB-list per a teacher answer is made. For example, if three teacher's answers are given, three PaMB-lists are created.

Some words have synonyms that have the same meaning. When writing answer, the student may use its synonym instead of the word in the teacher's correct answer. Even in this case, student's answer should be accepted as correct answer. To deal with this case, the PaMB-list generation module replaces the noun morphemes in the PaMB with the cluster representative term (CRT) of the morpheme using a synonym dictionary.

The synonym dictionary is constructed based on words presented in high school computer textbooks. We expanded the synonym dictionary with words not appearing in the textbook using the Word2vec model (using Korean Wikipedia).

Table 1 shows an example of a synonym dictionary constructed in this study.

### 3.2 Weighted PaMB-lists generation

The student's answer is scored through a similarity comparison with the correct answer presented by teacher. If the

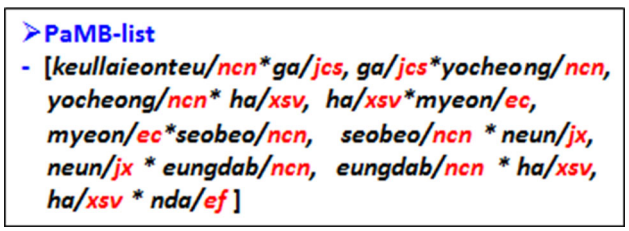


**Fig. 4** An example of a PaMB-list

**Table 1** An example of a synonym dictionary represented in English

| CRT | Terms |
|---|---|
| Keullaieonteu (client) | Keullaieonteu (client) |
| | Aepeulikeisheon (application) |
| | Teomineol (terminal) |
| | Beuraujeo (browser) |

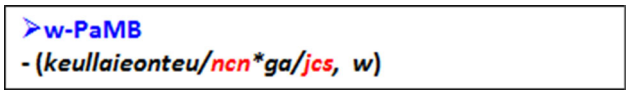


**Fig. 5** An example of a w-PaMB

student's answer matches the correct answer, he will get a full score, but if his matches partially, he will receive a partial score. If the student's answer is partially matched with the correct answer, grader determines how important the matching part is in the correct answer and assigns a partial score according to degree of the importance. In order to evaluate the significance of the parts that match the correct answer and the student answer for the partial scoring, we use a method to evaluate the importance of each PaMB constituting the correct answer, which is based on the frequency-based weighting mechanism.

The teacher can also present two or more correct answers for one question. In this case, the correct answers means different expressions for the same topic. For example, on the question 'Describe the role of client and server on the Web.' the teacher can register several correct answers:

- If a client requests a job from the server, the server replies the result to the client.
- If the client asks the server to solve the problem, the server solves the problem and presents the result to the client.
- If a client sends a request, the server processes the request and responds to the client.

The above example indicates that the same word can appear in multiple correct answers. In the correct answers, 'client' appears 6 times, 'server' 5 times, 'request' 3 times, and 'respond' totally 1 time, respectively. If the same word appears in multiple correct answers, it can be seen that the word contributes more than other words in expressing the correct answer.

The definition of weighted PaMB is as follows:

**Definition 4** *Weighted PaMB* (w-PaMB). A w-PaMB is a tuple consisting of a PaMB and its weight (Fig. 5).

The weight of a $\text{PaMB}_k$ is calculated in Eq. (1).

$$w\,(\text{PaMB}, d) = \alpha + (1-\alpha) \cdot \frac{f_{\text{PaMB},d}}{\max\limits_{\{\text{PaMB}' \in d\}} f_{\text{PaMB}',d}}, \qquad (1)$$

where $d$ is a set of all PaMBs contained in PaMB-lists (see Fig. 6). The $f_{\text{PaMB},d}$ is the occurrence frequency of PaMB contained in $d$. The $\alpha$ is a parameter to compensate for the difference between the smallest weight and the largest weight.

The weight is calculated based on a set of all PaMBs included in the three PaMB-lists in Fig. 6. The weight of

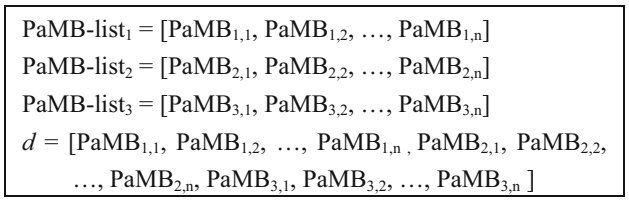


**Fig. 6** An integration of three PaMB-lists for computation of a PaMB weight

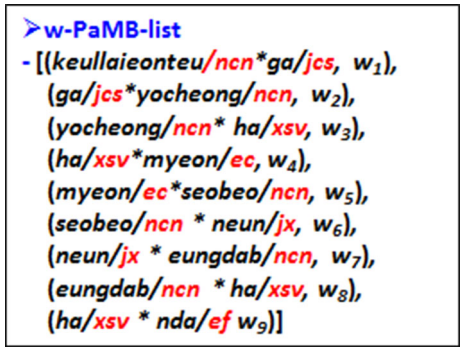


**Fig. 7** An example of a w-PaMB-list

a PaMB calculated in Eq. (1) is paired with the PaMB to form the w-PaMB.

When calculation of the weights of all PaMBs are completed, the PaMB-lists are converted into w-PaMB -lists.

**Definition 5** w-PaMB-List (w-PaMB-list). w-PaMB-list is an ordered sequence of a given w-PaMBs.

When we generate a w-PaMB-list, we use threshold $\theta$ to remove $\text{w-PaMB}_k$ with a weight less than $\theta$ ratio for efficient scoring. By removing the less important PaMBs to express the correct answer, it is possible to perform the scoring of the student answer more efficiently (Fig. 7).

### 3.3 LSPs generation

In this subsection, we present a method to generate *lexical semantic pattern* (LSP) from each w-PaMB-list.

LSP is designed to reflect the following characteristics of Korean. (i) The inversion of word (eojeol in Korean) in a sentence occurs relatively frequently compared to other languages. (ii) A word is usually represented through a biding of noun/or predicate morphemes and affix morpheme. The morpheme of postposition in a sentence may be different. But there are cases where the role of a word in a sentence is the same.

For example, the morpheme analysis results of '*keullaieonteu-ga*'' and '*keullaieonteu-neun*' are '*keullaieonteu/ncn*-ga/jcs*' and '*keullaieonteu/ncn*neun/jcs*'. '*ga*' and '*neun*' located behind the subject '*keullaieonteu*' are different morphemes. However, both of these morphemes have the same

role as postposition indicating the qualification of the subject after the substantive. Hence, we define LSP as a set to reflect these characteristics of Korean and the affix morphemes with same role is represented by POS in LSP tuple.

**Definition 6** *Lexical Semantic Pattern* (LSP). Given a w-PaMB-list = [w-PaMB$_1$, w-PaMB$_2$, …, w-PaMB$_n$], a LSP is a set of tuples $= \{t_1, t_2, \ldots, t_n\}$, where a tuple $t_i$ is a triple $(f_i, s_i, w_i)$, where

$$f_i = \begin{cases} FM_i & \text{if } FP_i \notin \text{ETC}; \\ FP_i & \text{otherwise.} \end{cases}, \quad s_i = \begin{cases} BM_i & \text{if } BP_i \notin \text{ETC}; \\ BP_i & \text{otherwise.} \end{cases}$$

and $w_i$ is w-PaMB$_i$.weight,

where $FM_i$ is w-PaMB$_i$.front.morp, $FP_i$ is PaMB$_i$.front.pos, $BM_i$ is w-PaMB$_i$.back.morp, $BP_i$ is PaMB$_i$.back.pos and *ETC* is a set of all tags except noun, verb and adjective tag.

Algorithm 1 describes the method for transforming w-PaMB-lists into *LSPs*. This algorithm takes a set *WP* of w-PaMB-lists, a set *ETC* of all tags except noun, verb and adjective tag as input.

It first initializes two sets *R* and *LSP* (line 1) to the empty set and then transforms into *LSP* (lines 3–20) for each w-PaMB-list *wp* in *WP* (line 2).

**Algorithm 1:** Transforming w-PaMB-lists into LSPs

**Input**: a set *WP* of w-PaMB-lists, a set *ETC* of all tags except noun, verb and adjective tag
**Output**: a set *R* of *LSPs*

```
1.  R ← ∅;  LSP ← ∅;
2.  for each w-PaMB-list wp in WP
3.    wpb ← first w-PaMB in wp;
4.    while (wpb != NULL) do
5.      first, second ← NULL;   weight ← 0;
6.      if (wpb.front.pos ∉ ETC)
7.          first ← wpb.front.morp;
8.      else
9.          first ← wpb.front.pos;
10.     if (wpb.back.pos ∉ ETC)
11.         second ← wpb.back.morp;
12.     else
13.         second ← wpb.back.pos;
14.     weight ← wpb.weight;
15.     t ← makelsptuple(first, second, weight);
16.     LSP ← LSP ∪ {t};
17.     wpb ← next w-PaMB in wp;
18.   end while
19.   R ← R ∪ {LSP};
20.   LSP ← ∅;
21. end for
22. return R;
```

It finds the first w-PaMB in *wp* and assigns it to *wpd* (line 3). Until *wpd* is NULL, it repeats transforming *wpd* (w-PaMB) into tuple of LSP (lines 4–18).

The details of transforming *wpd* (w-PaMB) into tuple of LSP are as follows:

It first initializes *first*, *second* and *weight* that make up one tuple in the *LSP* (line 5). Then, it checks whether *wpd.front.pos* is a noun or verb or an adjective (line 6). If the answer is true, it assign *wpd.front.morp* to *first* (line 7). Otherwise, it assigns *wpd.front.pos* to *first* (line 9). Also, it checks whether *wpd.back.pos* is a noun or verb or an adjective (line 10). If the answer is true, it assigns *wpd.back.morp* to *second* (line 11). Otherwise, it assigns *wpd.back.pos* to *second* (line 13). Finally, *weight* is assigned to *wpd.weight*, and the tuple *t* is made of these three values (lines 14–15). The *t* is added as an element of the set *LSP*, and we assign *wpd* the next w-PaMB in *wp* (lines 16–17).

Once an LSP is created, it adds the generated *LSP* to the set *R* and initialize the set *LSP* to an empty set (lines 19–20).

Finally, after the for-loop, the algorithm returns *R* and terminates (line 22).

### 3.4 Scoring

Finally, we present an algorithm that automatically scores student answers using the LSPs of teacher's answers generated in Sect. 3.3.

To score the student's answer, we first generate the PaMB-list from a student's answer (introduced in Sect. 3.1). Then, the student answer (PaMB-list) is scored using LSPs of teacher's answer.

The basic principles for scoring student's answer are as follows.

For each correct answer LSP, it compares with PaMB-list of the student's answer to see whether the PaMBs corresponding to each tuple for all tuples in the LSP are present in PaMB-list. If so, the student will receive the max score and terminates the scoring job. If it exists only partially, the sum of weights of the matched LSP tuples is preserved, and when there are no matching LSPs, the largest value among the sums of weights is used to calculate the partial score of the student answer.

**Algorithm 2:** Scoring of student's answer

**Input**: a PaMB-list *p* of the student, a set *LSPs* of the teacher, a max score *msc* for the question
**Output**: a score *sc* of the student

```
1.  sc, total_weight ← 0;
2.  for each LSP in LSPs
3.    mt ← 0; // the number of matched tuples
4.    lw ← 0; // weight of LSP
5.    for each tuple t in LSP
6.      pb ← first PaMB in p;
7.      while (pb != NULL and sc != msc)
8.        if (t.first is not POS)
9.          if (t.first = pb.front.morp)
10.           if (t.second is not POS)
11.             if (t.second = pb.back.morp)
12.               lw ← lw + t.weight;
13.               mt ← mt + 1;
14.           else if (t.second = pb.back.pos)
15.               lw ← lw + t.weight;
16.               mt ← mt + 1;
17.       else if (t.first = pb.front.pos)
18.         if (t.second is not POS)
19.           if (t.second = pb.back.morp)
20.               lw ← lw + t.weight;
21.               mt ← mt + 1;
22.         else if (t.second = pb.back.pos)
23.               lw ← lw + t.weight;
24.               mt ← mt + 1;
25.       pb ← next PaMB in p;
26.     end while
27.   end for
28.   if (mt = |LSP|)  // check perfect answer
29.     sc ← msc;   break;
30.   else if (lw > total_weight)
31.     total_weight ← lw;
32. end for
33. if (sc != msc) // calculate partial score
34.   normal ← max_{∀LSP∈L} (sum_{∀t∈LSP} (t.weight));
35.   sc ← (total_weight/normal)*msc;
36. return sc;
```

Algorithm 2 shows the detailed procedure for scoring student's answer. This procedure takes a PaMB-list *p* of the student, a set *LSPs* of the teacher, a max score *msc* for the question as input. First, it sets *sc* and *total_weight* to zero, where *sc* is a score of the student for the result, and *total_weight* is the largest weight of the student's answer for calculating partial score (line 1). Then, it processes the following for each *LSP* in *LSPs* (line 2).

It initializes *mt* and *lw* to be zero, where *mt* is the number of matched tuples and *lw* is a weight of *LSP* (lines 3-4). Then, it traverses the list *p* sequentially for each tuple *t* in *LSP*. If all PaMBs in the list *p* are traversed or the current score is *msc*, the list traversal ends. Otherwise, it checks whether the pattern of *first* and *second* of tuple *t* matches the pattern of *front* and *back* of PaMB *pb*, respectively, and it updates *lw* and *mt* if the patterns match (lines 5–27). After iterating over all the tuples in the *LSP*, it checks that they are the perfect answer (line 28). If condition is true, *sc* becomes *msc* and *sc* is returned as a result (lines 29, 36). Otherwise, it checks whether *lw* is greater than *total_weight*, and whether condition is true, *total_weight* is updated to *lw* (lines 30–31).

Finally, it modifies the current *sc* to a normalized subscore for answers that are not perfect. The modified *sc* is finally returned as a result (lines 33–36).

**Table 2** Pearson's correlation coefficient for LM and EM

| | Proposed automatic scoring system | Existing automatic scoring system |
|---|---|---|
| Teacher scoring | 0.754 | 0.617 |

## 4 Experiments and results

### 4.1 Data sets

In order to evaluate the proposed automatic scoring system, we conducted a test for 88 students at P girl high school in Korea. The teacher makes three web programming questions and makes two correct answers for each question. The teacher's answers are converted to LSPs by Algorithm 1. Sentences answered by students are converted into PaMB-lists and scored through Algorithm 2.

### 4.2 Comparisons of existing evaluation system

To evaluate the performance of the proposed automatic scoring method, we first compare two methods (LM and EM). The LM is LSP-based automatic scoring method, which is proposed in this paper, and the EM is existing automatic scoring method which uses the predicate normalization method that converts the verbs and adjectives of answers to noun forms (Bae 2013). We show our excellence through the correlation coefficient with the teacher scoring method (TM) for LM and EM, respectively.

Table 2 shows the Pearson's correlation coefficient for the LM and EM. The experiment results in Table 2 show that the correlation coefficient of LM and TM is high, i.e., 0.754. It was confirmed that the LM distinguishes correct answers from wrong answers more accurately than the EM does ($p < .01$).

### 4.3 Evaluation of synonym dictionary

In this subsection, we compare the accuracy to examine how the synonym dictionary affected the correct evaluation.

Precision, recall and F-measure are often used as evaluation criteria in measuring performance of system. To get

precision, recall and F-measure, a classifier evaluation metric is used like Table 3.

Using a classifier evaluation metric, precision, recall and F-measure can be calculated as Eqs. (2)–(4).

$$\text{precision} = \frac{a}{a+c} \tag{2}$$

$$\text{recall} = \frac{a}{a+b} \tag{3}$$

$$\text{F-measure} = 2 \cdot \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{4}$$

The teacher evaluates the student answer as correct or incorrect; on the other hand, the system evaluates student answers in a way that gives partial scores. In order to measure precision, recall and F-measure, the system has to evaluate student answer as correct or incorrect like Table 3. If the difference between the max score of the question and the score evaluated by the system is less than or equal to $\varepsilon$, we classify the answer as correct, otherwise classify as incorrect.

We obtain results evaluated by system in two methods. The first method (FM) does not use the synonym dictionary when comparing LSPs to student answers. In this case, if students do not write the same morpheme in the teacher's answer, it will not be evaluated correctly. The second method (SM) uses the synonym dictionary when comparing LSPs and student answers. In this case, if students write the synonym of morpheme in the teacher's answer, it will be evaluated correctly.

Figure 8 shows the comparison of the results of precision, recall and F-measure by two methods (FM and SM). In all three questions, the SM is found to be higher precision, recall and F-measure than the FM. F-measure values for three questions are 0.63, 0.66 and 0.51 for FM and 0.71, 0.76 and 0.80 for SM, respectively. As a result, the SM is superior to the FM.

**Table 3** A classifier evaluation metric for precision and recall and F-measures

| | | The number of student answers evaluated by the system | |
|---|---|---|---|
| | | Correct | Incorrect |
| The number of student answers evaluated by the teacher | Correct | a | b |
| | Incorrect | c | d |

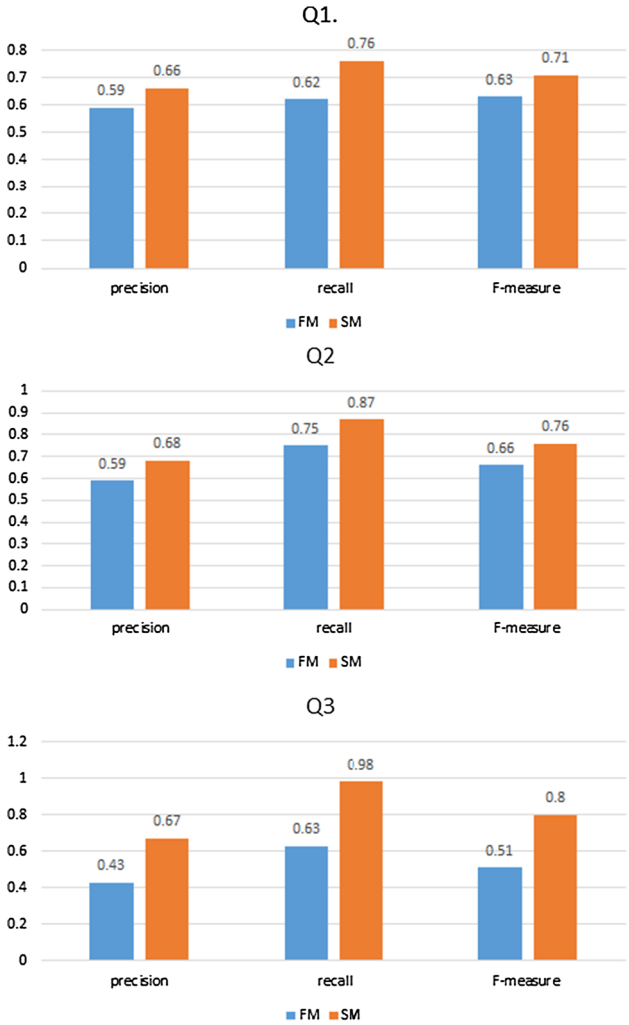


**Fig. 8** Results precision, recall, F-measure of three questions

## 5 Conclusion

In this paper, we developed an automatic scoring system that improves the accuracy of descriptive evaluation by reflecting Korean characteristics. In order to reflect the feature of Korean, we defined PaM, PaMB, w-PaMB and LSP. Also, we proposed an algorithm to generate LSP from each w-PaMB-list and an algorithm that automatically scores student answers using the LSPs. The proposed method in this paper can be applied easily to other agglutinative languages like Korean. In order to improve the accuracy of the descriptive answer evaluation, the system expanded the correct morpheme by using the synonym dictionary.

Experiments showed that our system has a higher correlation with teacher scoring scores than existing automatic scoring systems. Experimental results presented that the evaluation method of morpheme and postposition combined unit is more similar to the teacher evaluation result than the existing noun unit evaluation method. It is shown that evaluating

the morpheme of PaMB by changing the vocabulary to CRT through the synonym dictionary increases the accuracy of evaluation.

Future research is to investigate whether the proposed synonym dictionary-based LSP scoring method is not restricted to computer subject, but similar accuracy is obtained in various subjects.

**Acknowledgements** This research was supported by the Special Research Fund of College of Education, Korea University in 2013.

**Compliance with ethical standards**

**Conflict of interest** The authors declare that there is no conflict of interests.

## Appendix: POS tag set

| | Clasificar | Subclass | |
|---|---|---|---|
| Uninflected word | Noun | Nondeclarative noun | ncn |
| | | Predicative movement noun | ncpa |
| | | Predicative state noun | ncps |
| | | Proper noun | nq |
| | | Non-measure dependent noun | nbn |
| | | Measure dependent noun | nbu |
| | Pronoun | | np |
| | Rhetoric | | nng |
| Predicate | Verb | | pv |
| | Adjective | | pa |
| | Auxiliary predicate | | px |
| Flourish word | Determiner | General determiner | mmg |
| | | Number determiner | mmc |
| | Adverb | General adverb | Mag |
| | | Conjunctive adverb | mac |
| Independence word | Exclamation | | i |
| Relative word | Postposition | Subjective postposition | jcs |
| | | Prenominal postposition | jcm |
| | | Objective postposition | jco |
| | | Adverbial postposition | jca |
| | | Auxiliary postposition | Jx |
| | | Conjunctive postposition | jcj |
| Dependence form | Ending | Prefinal ending | ep |
| | | Final ending | ef |
| | | Connective ending | ec |
| | | Auxiliary connective ending | ecx |
| | | Noun changing ending | etn |
| | | Adjective changing ending | etm |
| | Suffix | Noun derivation suffix | xsn |
| | | Verb derivation suffix | xsv |
| | | Adjective derivation suffix | xsm |
| | | Adverb derivation suffix | xsa |

## References

Burrows S, Gurevych I, Stein B (2015) The eras and trends of automatic short answer grading. Int J Artif Intell Educ 25(1):60–117

Burstein J, Chodorow M (1999) Automated essay scoring for nonnative English speakers. In: Proceeding ASSESSEVALNLP '99 proceedings of a symposium on computer mediated language assessment and evaluation in natural language processing, pp 68–75

Bae B-G (2013) Automatic-scoring method for Korean free-text answer through predicate normalization. M.A. dissertation, KookMin University

Callear D, Jerrams-Smith J, Soh V (2001) CAA of short non-MCQ answers. In: Proceeding of the 5th international CAA conference

Chakrabartty S, Cauwenberghs G (2007) Gini support vector machine: quadratic entropy based robust multi-class probability regression. J Mach Learn Res 8:813–839

Cho W, Oh J, Lee J, Kim Y-S (2005) An intelligent marking system based on semantic kernel and Korean WordNet. KIPS Trans Part A 12A(6):539–546

Clariana RB, Wallace P (2007) A computer-based approach for deriving and measuring individual and team knowledge structure from essay questions. J Educ Comput Res 37(3):211–227

Cruse DA (1986) Lexical semantics. Cambridge University Press, Cambridge

Dobrov BV, Loukachevitch NV, Yudina TN (1997) Conceptual indexing using thematic representation of texts. In: TREC-6, pp 403–413

ETRI (2013) ETRI Korean phoneme converter. Speech Language Information Research Department

Jacobs PS, Krupka GR, Rau LF (1991) Lexico-semantic pattern matching as a companion to parsing in text understanding. In: Proceeding HLT '91 proceedings of the workshop on speech and natural language, pp 337–341

Jadidinejad AH, Mahmoudi F (2014) Unsupervised short answer grading using spreading activation over an associative network of concepts. Can J Inf Library Sci 38:287–303

Jang ES, Kang SS, Noh EH, Kim MH, Sung KH, Seong TJ (2014) KASS: Korean automatic scoring system for short-answer questions. In: Proceedings of the 6th international conference on computer supported education: CSEDU, Barcelona, Spain, vol 2, pp 226–230

Kim JE, Chae JM, Jung SY (2014) Automatic scoring system based on Korean lexico-semantic pattern. Int J Appl Eng Res 9(22):14499–14510

Loukachevitch NV, Dobrov BV (2002) Evaluation of thesaurus on sociopolitical life as information-retrieval tool. In: Gonzalez Rodriguez M, Paz Suarez Araujo C (eds) The third international conference on linguistic resources and evaluation (LREC-2002), Gran Canaria, Spain, vol 1

Valenti S, Neri F, Cucchiarelli R (2003) An overview of current research on automated essay grading. J Inf Technol Educ 2:319–330