

A Study of Distributed Semantic Representations for Automated Essay Scoring

Cancan Jin, Ben He^(✉), and Jungang Xu^(✉)

School of Computer and Control Engineering,
University of Chinese Academy of Sciences,
Geese-resting Lake Campus, Beijing 101408, China
jincancan15@mailsucas.ac.cn, {benhe,xujg}@ucas.ac.cn

Abstract. Automated essay scoring (AES) applies machine learning and NLP techniques to automatically rate essays written in an educational setting, by which the workload of human raters is considerably reduced. Current AES systems utilize common text features such as essay length, *tf-idf* weight, and the number of grammar errors to learn a scoring function. Despite the effectiveness brought by those common features, the semantics within the essay text is not well considered. To this end, this paper presents a study of the usefulness of the distributed semantic representations to AES. Novel features based on word or paragraph embeddings are combined with the common text features in order to improve the effectiveness of the AES systems. Evaluation results show that the use of the distributed semantic representations are beneficial for the task of AES.

Keywords: Automated essay scoring · Distributed semantic representations · Embeddings

1 Introduction

Automated essay scoring (AES) is usually considered as a machine learning problem [1–3] where learning algorithms such as K-nearest neighbor and support vector machines for ranking are applied to learn a rating model for a given essay prompt with a set of training essays rated by human assessors [4]. Currently, the AES systems have been widely used in large-scale English writing tests, e.g. Graduate Record Examination (GRE), to reduce the human efforts in the writing assessments.

In general, existing AES systems are based on a number of common text features that are not linked to intuitive dimensions of semantics or writing quality, such as lexical complexity, grammar errors, syntactic complexity, organization and development, coherence, etc. However, these shallow text features are not able to represent the semantic content of essays, resulting in limited robustness and effectiveness [5].

Recently, the word level, phrase level, and sentence level semantic representations of documents are successfully applied to compute the syntactic and semantic similarity in quite a few natural language processing (NLP) tasks. For example, Tomas et al. propose a method based on representations of words and sentences that achieves promising results for movie rating prediction on a crawl of IMDB [8]. Richard et al. propose a method based on the continuous representations of sentences that obtains good performance on the Stanford background dataset [9]. There are also efforts in developing methods for extracting the semantic representations of documents [9–11]. For instance, a simple approach is to use a weighted average of all word vectors in the document [12]. A more sophisticated approach is to learn continuous distributed vector representations for pieces of texts [13]. For the task AES, there has been little success of the application of the semantic features as far as we are aware of.

To this end, this paper presents an investigation in the usefulness of various novel features in indicating the writing quality of essays. The new features are derived based on different approaches to generating distributed representations of words, paragraphs, and documents, including latent Dirichlet allocation (LDA) [17], Word2Vec [18], and PV-DBOW [13]. Experimental results on the publicly available dataset ASAP indicate that the new features based on the semantic similarity features and distributed semantic representations of essays achieve higher agreement with human raters than the use of only the common text features. In our evaluation, the use of the new features can achieve up to 12.33% improvement in Kappa, and 18.61% improvement in nRMSE against the baseline.

2 Common Text Features

This section introduces common text features widely used in the previous AES methods [1, 7, 14, 15], which are listed in Table 1. The detailed description of the features is given below.

- *Statistics of word length*: The *mean* and *variance* of word length in characters. These can be indicators for the degree of complexity a writer can master since the unusual words tend to be longer. The number of unique words appeared in an essay, normalized by the essay length in words.
- *Statistics of sentence length*: The *mean* and *variance* of sentence length in words. The variety of the length of sentences potentially reflects the complexity of syntactics.
- *Statistics of essay length*: The essay length is measured by the number of words and the number of characters in an essay. Essays are usually written under a time limit, so the essay length can be a useful predictor of the productivity of the writer. The fourth root of essay length in words is proved to be highly correlated with the essay score [15].
- *Clauses*: The *mean* number of clauses in each sentence, normalized by the number of sentence in an essay. The *maximum* number of clauses of a sentence in an essay. The *mean length* of sentences that contain at least one clause.

Table 1. Common text features.

No.	Feature
1	Mean and variance word length in characters
2	Mean length of clauses
3	Essay length in characters and words
4	Number of spelling errors
5	The number of prepositions and commas
6	Mean number of clauses per sentence
7	Mean and variance of sentence length in words
8	Maximum number of clauses of a sentence
9	Semantic vector similarity based on <i>LSA</i>
10	Mean cosine similarity of word vectors by <i>tf-idf</i>
11	The average height of the parser tree of each sentence in an essay
12	Word bigram/trigram frequency <i>tf</i> divided by collection frequency <i>TF</i>
13	POS bigram/trigram frequency <i>tf</i> divided by collection frequency <i>TF</i>

- *Sentence structure*: The number of prepositions and commas in each sentence, normalized by words in sentences. The average height of the parser tree of each sentence in an essay. The average of the sum of the depth of all nodes in a parser tree of each sentence in an essay. The more complicated the sentences are, the higher complexity the parser trees exhibit. It is therefore necessary to utilize the sentence structure to indicate the essay quality.
- *Spelling errors*: Grammatical or spelling errors are one of the most obvious indicators of bad essays, which are detected by the spelling check API provided by LanguageTool¹.
- *Word bigram and trigram*: The level of grammar and fluency of an essay can be measured by the mean *tf/TF* of word bigrams and trigrams [16] (*tf* is the frequency of bigram/trigram in a single essay and *TF* is the frequency of bigram/trigram in the whole essay collection). We assume a bigram or trigram with high *tf/TF* as a grammar error because high *tf/TF* means that this kind of bigram or trigram is not commonly used in the whole essay collection but appears in the specific essay.
- *POS bigram and trigram*: Mean *tf/TF* of POS bigrams and trigrams. The Part-of-Speech tagging of each word is done by the Stanford Parser².
- *Word vector similarity*: Mean cosine similarity of word vectors, in which the element is the term frequency multiplied by inverse document frequency (*tf-idf*) of each word. It is calculated as the weighted mean of cosine similarities and the weight is set as the corresponding essay score.

¹ <https://www.languagetool.org>.

² <http://nlp.stanford.edu/software/corenlp.shtml>.

- *Semantic vector similarity*: Semantic vectors are generated by Latent Semantic Analysis [6]. The calculation of mean cosine similarity of semantic vectors is the same with word vector similarity.

Each feature is normalized to be within $[0, 1]$. The features introduced in this section include most of the common text features used in recent studies on AES, which lead to state-of-the-art results [1, 4, 7, 16, 20]. Therefore, the AES system trained by those common text features is used as the baseline in this paper.

3 Semantic Representations for AES

This section introduces the semantic features involved in this study. Section 3.1 introduces the methods used for learning the semantic representations of essays, from which the semantic features are generated, as in Sect. 3.2.

3.1 Methods for Vector Representations

Other than the previously applied LSA approach in [7], we propose to generate semantic features based on the following recent methods for the vector representations. A brief introduction of how to obtain semantic embeddings of essays through these learning algorithms is given below.

Latent Dirichlet Allocation (LDA) is a generative probabilistic model of a corpus [17]. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. The probabilistic distribution on all topics of a document is considered as a kind of semantic representation of a document. Using LDA, the i th dimension of the semantic representation of the essay is given by the probability that the essay belongs to topic i . To analyze the effectiveness of the different number of topics, the number of topics is set as 5, 6, 7, 8, 9, 10, respectively. Those settings are found to be the most effective in the preliminary experiments. Results obtained using the different settings above are also presented in Sect. 5.2.

Continuous Skip-gram Model is used to learn high-quality distributed vector representations that capture syntactic and semantic relationships between words [18]. In continuous skip-gram model, every word is mapped to a unique vector, and all the word vectors are stacked in a word embedding matrix W generated by the model. The weighted mean of word embeddings of words appeared in an essay is used as a semantic representation of the essay, which the weight is set as the tf/TF . The i th dimension of the semantic representation of the essay is given by:

$$word.Vec_i = \frac{\sum_{j=1}^n weight_j * W_{ji}}{n} \quad (1)$$

where n is the number of unique words in an essay, $weight_j$ is the tf/TF of the j th word in the essay, and W_{ji} is the i th dimension of the word vector of the j th word in the essay.

In this paper, we use two different datasets to learn the word embeddings, one is the publicly available ASAP dataset (see Sect. 4.1 for details), and the other is the GoogleNews dataset³. The latter is a large-scale news corpus which may lead to better training of word embeddings. The word embeddings obtained on ASAP is trained by Word2Vec, and the dimension of word embedding is set as 50, 100, 200 and 300, respectively. Those settings are found to be the most effective in the preliminary experiments. Results obtained using the different settings above are also presented in Sect. 5.2. Using GoogleNews, the word vectors are pre-trained on 100 billion words of Google news dataset and are of length 300.

Distributed Bag of Words Version of Paragraph Vectors Model: Tomas et al. propose the distributed bag of words version of paragraph vector model (PV-DBOW) [13]. The PV-DBOW model learns the paragraph vector based on the continuous skip-gram model. A notable difference between the outcome of the PV-DBOW model and the continuous skip-gram model is that the PV-DBOW model generates the vector representations of paragraphs, in addition to the word vectors. The paragraph vector representations are obtained by the PV-DBOW model trained on the ASAP dataset. GoogleNews is not used as it only comes with word embeddings. The same as the word embeddings, the dimension of paragraph embedding is set to 50, 100, 200, and 300, respectively, for effectiveness reason.

3.2 Semantic Features

In this paper, we present two ways to using the semantic representations of essays for generating the semantic features for AES, namely Vector Similarity and Dimension Extension.

Vector Similarity: It is calculated as the mean of all weighted cosine similarities between the given essay and the other essays for a given prompt. Assuming w_1, w_2, \dots, w_m are the semantic representations of essays in the specific essay set, Sim_i is the Vector Similarity of the i th essay:

$$Sim_i = \frac{\sum_{j=1, j \neq i}^m r_j \cdot \frac{\vec{w}_i \cdot \vec{w}_j}{\|\vec{w}_i\| \times \|\vec{w}_j\|}}{(m-1) \cdot \sum_{j=1, j \neq i}^m r_j} \quad (2)$$

where m is the number of essays associated to the given prompt, and r_j is the actual rating of the j th essay. Using Vector Similarity, only a single semantic feature is generated from the essay embeddings.

Dimension Extension: The feature vector of a given essay is extended by the entire semantic representations of the essay. Each dimension of the essay embedding is regarded as a semantic feature of the essay. In other words, the size of the feature vector of the given essay is extended by the number of dimensions of the entire semantic representations of the essay.

³ <https://drive.google.com/file/d/0B7XkCwpI5KDYNINUTTISS21pQmM/edit?usp=sharing>.

Table 2. Semantic features.

Semantic features	Feature description
lda_sim_k	Vector Similarity feature learned by LDA The number of topics k is set as 5, 6, 7, 8, 9, 10, respectively
lda_vec_k	Dimension Extension feature learned by LDA The number of topics k is set as 5, 6, 7, 8, 9, 10, respectively
word_sim_k	Vector Similarity feature learned from ASAP by skip-gram The number of dimensions k is set as 50, 100, 200, 300, respectively
word_vec_k	Dimension Extension feature learned from ASAP by skip-gram The number of dimensions k is set as 50, 100, 200, 300, respectively
google_sim_300	Vector Similarity feature learned from GoogleNews The dimension of the word embeddings is 300
google_vec_300	Dimension Extension feature learned from GoogleNews The dimension of the word embeddings is 300
para_sim_k	Vector Similarity feature learned from ASAP by PV-DBOW The number of dimensions k is set as 50, 100, 200, 300, respectively
para_vec_k	Dimension Extension feature learned from ASAP by PV-DBOW The number of dimensions k is set as 50, 100, 200, 300, respectively

We can generate a list of semantic features through the above methods on the basis of semantic representations of essays introduced in Sect. 3.1.

A list of the semantic features used in this paper is summarized in Table 2. For example, when the *para_vec_100* feature is used, the essay feature vector is extended by the semantic paragraph vector with 100 dimensions. Instead, if *para_sim_100* is used, the learned paragraph embeddings have 100 dimensions, and the essay feature vector is extended by a single dimension, which is the similarity between the semantic paragraph vector of essay and other essays in the same essay set.

4 Experimental Settings

This section presents our experimental design, including the dataset used, the evaluation metrics of the AES system, and the learning algorithms.

4.1 Dataset

The dataset used in our experiments comes from the Automated Student Assessment Prize (ASAP)⁴. Dataset in this competition consists of eight essay sets. Each essay set was generated from a specific prompt. All essays received a resolved score, namely the actual rating, from professional human raters. As the official test data is no longer available, the evaluation is done by 10-fold cross-validation on the training data, split by random partitioning.

⁴ <https://www.kaggle.com/c/asap-aes/data>.

4.2 Evaluation Metrics and Learning Algorithms

In this paper, we use Kappa, Pearson correlation coefficient, Spearman correlation coefficient and normalized root-mean-squared error to evaluate the agreement between the ratings given by the AES system and the actual ratings. They are widely accepted as reasonable evaluation measures for the AES systems [14, 19, 20].

Quadratic Weighted Kappa is a statistical metric which is used to measure inter-rater agreement. Quadratic weighted Kappa takes the degree of disagreement between raters into account. The kappa metric is computed by the mean of the kappa values across all essay sets after applying the Fisher Transformation⁵, instead of the average of the raw kappa values over all essay sets.

Pearson Correlation Coefficient [21] is used to measure the strength of a linear association between two variables.

Spearman Correlation Coefficient [22] assesses how well the relationship between two variables can be described using a monotonic function.

In addition to the above three human-machine agreement metrics, the **Normalized root-mean-squared error** (nRMSE) [23] measures the prediction error of the essay ratings. The ratings of a given essay topic are normalized to be within $[0, 1]$ such that the errors among different prompts are comparable. *Different from the other three metrics, a lower nRMSE value indicates better effectiveness.* The nRMSE reported in the results is averaged over all test essays in the whole dataset. All statistical tests are based on Analysis of Variance (ANOVA).

In this paper, we use K-nearest neighbor (KNN) and support vector machines for ranking (SVM-rank) to predict ratings of essays. These two classical algorithms are widely used in recent studies on the AES systems [14, 15].

K-nearest Neighbors (KNN) [24] is a classical classification algorithm commonly used in automated essay scoring. Using KNN, we select the K essays in the training collection that are most similar to the test essay. Then the predicted score of the test essay is the average of the scores of the K essays. The parameter K is set by grid search on the ASAP validation set.

For **SVM-rank**, the linear kernel function is used in the experiments. The parameter C , which controls the trade-off between empirical loss and regularizer, is set by grid search on the ASAP validation set. To determine the final rating of a given essay, we take the average rating of k essays whose scores are closest to the given essay. The parameter k is also set by grid search on the ASAP validation set. We use the implementation of SVM-rank in SVMrank package⁶.

5 Evaluation Design and Results

5.1 Evaluation Design

In order to examine the effectiveness of the semantic features when applied to AES, the experiments conducted in this paper are organized as follows.

⁵ <https://www.kaggle.com/c/asap-aes/details/evaluation>.

⁶ <http://svmlight.joachims.org/>.

Table 3. Performance of the semantic features generated by Word2Vec with different numbers of dimensions.

Methods	Vector similarity						
	Metrics	Base	word _sim_50	word _sim_100	word _sim_200	word _sim_300	google _sim_300
SVM_rank	Kappa	.7423	.7656	.7716	.7600	.7695	.7592
	Pearson	.7793	.8115	.8189	.8082	.8109	.8079
	Spearman	.7355	.7702	.7797	.7749	.7722	.7678
	nRMSE	.1709	.1532	.1486	.1533	.1530	.1571
KNN	Kappa	.7103	.7728	.7746	.7727	.7734	.7754
	Pearson	.7429	.7890	.7881	.7890	.7845	.7928
	Spearman	.7225	.7768	.7766	.7746	.7765	.7781
	nRMSE	.1811	.1650	.1648	.1649	.1655	.1638
Methods	Dimension extension						
	Metrics	Base	word _vec_50	word _vec_100	word _vec_200	word _vec_300	google _vec_300
SVM_rank	Kappa	.7423	.7895	.7784	.7863	.7766	.7817
	Pearson	.7793	.8292	.8215	.8242	.8018	.8169
	Spearman	.7355	.7912	.7812	.7862	.7850	.7817
	nRMSE	.1709	.1454	.1459	.1464	.1569	.1463
KNN	Kappa	.7103	.7190	.7028	.7182	.7114	.7365
	Pearson	.7429	.7492	.7428	.7421	.7501	.7678
	Spearman	.7225	.6950	.6905	.6995	.6978	.7396
	nRMSE	.1811	.1716	.1718	.1703	.1706	.1672

Effectiveness of the Individual Semantic Features with Different Numbers of Dimensions: To investigate the effectiveness of the semantic features with different numbers of dimensions, six sets of experiments are conducted. Each set of experiments corresponds to one specific semantic feature, in addition to the baseline that uses the common text features in Table 2, as presented in Tables 3, 4 and 5, respectively.

Comparison to the Baseline Using a Combination of the Best Individual Semantic Features Based on Vector Similarity and Dimension Extension: In these experiments, we compare the semantic features to the baseline that uses the common text features. The semantic features are *word_sim*, *word_vec*, *lda_sim*, *lda_vec*, *para_sim*, *para_vec* and *sim.best+vec.best*. Out of these semantic features with different embedding dimensions presented in Tables 3, 4 and 5, we choose the best individual semantic feature in each table to be evaluated against the baseline. *sim.best+vec.best* denotes the concatenation of the best Vector Similarity feature and the best Dimension Extension feature out of Tables 3, 4 and 5, which correspond to Word2Vec, PV-DBOW, and LDA, respectively.

Table 4. Performance of the semantic features generated by PV-DBOW with different numbers of dimensions.

Methods	Vector similarity					
	Metrics	Base	para_sim_50	para_sim_100	para_sim_200	para_sim_300
SVM_rank	Kappa	.7423	.7631	.7624	.7707	.7641
	Pearson	.7793	.8085	.8056	.8198	.8083
	Spearman	.7355	.7684	.7633	.7733	.7696
	nRMSE	.1709	.1549	.1543	.1523	.1535
KNN	Kappa	.7103	.7663	.7669	.7627	.7626
	Pearson	.7429	.7821	.7831	.7818	.7803
	Spearman	.7225	.7610	.7620	.7592	.7903
	nRMSE	.1811	.1648	.1646	.1649	.1650
Methods	Dimension extension					
	Metrics	Base	para_vec_50	para_vec_100	para_vec_200	para_vec_300
SVM_rank	Kappa	.7423	.7731	.7691	.7624	.7671
	Pearson	.7793	.8205	.8121	.8079	.8099
	Spearman	.7355	.7838	.7702	.7706	.7656
	nRMSE	.1709	.1498	.1545	.1544	.1545
KNN	Kappa	.7103	.7892	.7908	.7866	.7907
	Pearson	.7429	.8116	.8128	.8126	.8104
	Spearman	.7225	.7943	.7969	.7946	.7920
	nRMSE	.1811	.1625	.1617	.1620	.1621

Baseline uses all the common text features in Table 1 to learn a rating model for AES. The results are listed in Table 6. The last column in Table 6 presents the results of *sim_best+vec_best*.

Take KNN for example, *lda_sim/vec_6*, *google_sim/vec_300*, and *para_sim/vec_100* are compared with the baseline as they are the best out of the different settings of parameter *k*. *sim_best+vec_best* means that the feature set used is the concatenation of *Baseline*, *para_vec_100* and *google_sim_300*.

5.2 Evaluation Results

Firstly, the performance of the individual semantic features are evaluated. Tables 3, 4 and 5 present the evaluation results brought by the use of individual semantic features in addition to the common text features, with respect to different numbers of dimensions. Each of the tables corresponds to the results of the semantic features generated by a single learning method, i.e. Word2Vec, PV-DBOW, or LDA. In Tables 3, 4 and 5, the best result of each semantic feature is in **bold**.

According to Tables 3, 4 and 5, the effectiveness of the semantic features is in general stable with different numbers of embedding dimensions in different evaluation metrics. Therefore, changing this parameter setting does not have

Table 5. Performance of the semantic features generated by LDA with different numbers of dimensions.

Methods	Vector similarity							
	Metrics	Base	lda_sim_5	lda_sim_6	lda_sim_7	lda_sim_8	lda_sim_9	lda_sim_10
SVM_rank	Kappa	.7423	.7657	.7555	.7579	.7692	.7616	.7495
	Pearson	.7793	.8131	.8040	.8062	.8139	.8129	.7998
	Spearman	.7355	.7716	.7674	.7656	.7742	.7740	.7595
	nRMSE	.1709	.1535	.1561	.1551	.1525	.1531	.1561
KNN	Kappa	.7103	.7252	.7553	.7304	.7355	.7204	.7378
	Pearson	.7429	.7518	.7718	.7643	.7483	.7356	.7595
	Spearman	.7225	.7292	.7478	.7321	.7293	.7288	.7311
	nRMSE	.1811	.1713	.1650	.1675	.1696	.1695	.1685
Methods	Dimension extension							
	Metrics	Base	lda_vec_5	lda_vec_6	lda_vec_7	lda_vec_8	lda_vec_9	lda_vec_10
SVM_rank	Kappa	.7423	.7679	.7713	.7647	.7671	.7566	.7603
	Pearson	.7793	.8124	.8163	.8134	.8121	.8026	.8094
	Spearman	.7355	.7674	.7752	.7682	.7652	.7669	.7745
	nRMSE	.1709	.1525	.1523	.1533	.1528	.1541	.1540
KNN	Kappa	.7103	.6811	.7239	.7098	.6931	.6831	.6640
	Pearson	.7429	.7183	.7532	.7430	.7288	.7312	.7292
	Spearman	.7225	.7006	.7298	.7170	.7072	.7112	.7097
	nRMSE	.1811	.1851	.1708	.1781	.1845	.1801	.1809

a significant impact on the performance of the individual features. Moreover, according to Table 3, the word embeddings learned from ASAP appears to have slight better performance than those learned from GoogleNews when SVM-rank is used, and the other way around when KNN is used. In addition, comparing the evaluation results of using Vector Similarity and Dimension Extension of the same embeddings, we find no conclusive results. When SVM-rank is used, the Vector Similarity features have overall slightly better performance than the Dimension Extension features. However, when KNN is used, the Vector Similarity features have better performance when generated by Word2Vec (Table 3) and LDA (Table 5), while the Dimension Extension features gave better performance when generated by PV-DBOW. Such diverse results suggest the potential usefulness to combine the best individual semantic features based on Vector Similarity and Dimension Extension, respectively.

Next, the best Vector Similarity and Dimension Extension features are combined in order to make the best use of those semantic features. Table 6 compares the use of the combination of the best semantic features against the baseline. The last column in Table 6 presents the results of *sim_best+vec_best*, the concatenation of the baseline features, and the best features generated by Vector Similarity and Dimension Extension, respectively. A * indicates a statistically significant improvement over the baseline according to the ANOVA test. According to Table 6, all semantic features we present in this study have improvements over the baseline, and *sim_best+vec_best* has the best performance in all cases.

Table 6. Main evaluation result: best individual features (columns 3–8) against the baseline, and the combination (the last column) of the best Vector Similarity (sim) and Dimension Extension (vec) features against the baseline.

SVM_rank	Base	lda _sim_8	lda _vec_6	word _sim_100	word _vec_50	para _sim_200	para _vec_50	word _sim_100 + word _vec_50
Kappa	.7423	.7692 + 3.62%*	.7713 + 3.91%*	.7716 + 3.95%*	.7895 + 6.36%*	.7707 + 3.83%*	.7731 + 4.15%*	.8016 + 7.99%*
Pearson	.7793	.8139 + 4.44%*	.8163 + 4.75%*	.8189 + 5.08%*	.8292 + 6.40%*	.8198 + 5.20%*	.8205 + 5.29%*	.8374 + 7.46%*
Spearman	.7355	.7742 + 5.26%*	.7752 + 5.40%*	.7797 + 6.01%*	.7912 + 7.57%*	.7733 + 5.14%*	.7838 + 6.57%*	.8031 + 9.19%*
nRMSE	.1709	.1525 − 10.77%*	.1523 − 10.88%*	.1486 − 13.05%*	.1454 − 14.92%*	.1523 − 10.88%*	.1498 − 12.34%*	.1391 − 18.61%*
KNN	Base	lda _sim_6	lda _vec_6	google _sim_300	google _vec_300	para _sim_100	para _vec_100	google _sim_300 + para _vec_100
Kappa	.7103	.7553 + 6.34%*	.7239 + 1.91%	.7754 + 9.16%*	.7365 + 3.69%*	.76697.97%*	.7908 + 11.33%*	.7979 + 12.33%*
Pearson	.7429	.7718 + 3.89%*	.7532 + 1.39%	.7928 + 6.72%*	.7678 + 3.35%*	.7831 + 5.41%*	.8128 + 9.41%*	.8161 + 9.85%*
Spearman	.7225	.7478 + 3.50%*	.7298 + 1.01%	.7781 + 7.69%*	.7396 + 2.37%*	.7620 + 5.47%*	.7969 + 10.30%*	.8001 + 10.74%*
nRMSE	.1811	.1650 − 8.89%*	.1708 − 5.69%*	.1638 − 9.55%*	.1672 − 7.68%*	.1646 − 9.11%*	.1617 − 10.71%*	.1612 − 10.99%*

This shows that it is beneficial to combine the semantic features generated by both methods. When using SVM-rank, the features generated by Dimension Extension have overall better performance than those generated by Vector Similarity and the effectiveness of features generated by word embeddings outperform the features generated by PV-DBOW and LDA.

Using SVM-rank, the improvements brought by all semantic features generated by Vector Similarity and Dimension Extension are statistically significant when the effectiveness is measured by all four evaluation metrics. Using KNN, *google_sim_300* outperforms *para_sim_100* and *lda_sim_6*, and *para_vec_100* has better performance than *google_vec_300* and *lda_vec_6*. According to the ANOVA significance test, the improvements brought by *google_sim_300*, *google_vec_300*, *para_sim_100*, *para_vec_100*, *lda_sim_6* and *sim_best+vec_best* are statistically significant when the effectiveness is measured by Kappa, Pearson and Spearman. All improvements are statistically significant when the effectiveness is measured by nRMSE.

Overall, the results show that the use of the semantic features can indeed improve the effectiveness of AES on top of the common text features. As shown in Table 6, it is particularly encouraging that a combination of the best features can achieve up to 12.33% improvement in Kappa, and 18.61% improvement in nRMSE. Therefore, it is also recommended to combine the best features generated by Vector Similarity and Dimension Extension, in order to achieve the maximized performance of AES. It is widely accepted that the agreement between pro-

fessional human raters ranges from 0.70 to 0.80, measured by quadratic weighted Kappa or Pearson’s correlation [3]. In Table 6, the semantic features achieve a Kappa of 0.8016 and a Pearson’s correlation of 0.8374, suggesting their potential usefulness in automated essay scoring.

6 Conclusions and Future Work

In summary, this paper presents an investigation on the effectiveness of using semantic vector representations for the task of automated essay scoring (AES). According to the evaluation results on the standard ASAP English dataset, the effectiveness brought by our proposed semantic representations of essays depends on the learning algorithms and the evaluation metrics used. On the other hand, the effectiveness of individual semantic features is stable with respect to different numbers of dimensions. Results show that statistically significant improvement over the baseline can be achieved by applying our proposed semantic features listed in Table 2. Results also show that the concatenation of the best features generated by Vector Similarity and Dimension Extension, namely feature *sim_best+vec_best* has the best effectiveness among all features involved in this investigation. Moreover, the semantic features based on word embeddings lead to better effectiveness than those based on LDA embeddings and paragraph embeddings.

In the future, we plan to continue the research by mining effective features based on different sources of information, e.g. the structure of a given essay. We also plan to further improve this work by using the embeddings as input to a deep neural network, in order to learn an AES model.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (61472391).

References

1. Attali, Y., Burstein, J.: Automated essay scoring with e-rater. *J. Technol. Learn. Assess.* **4**(3), 7–15 (2006)
2. Dikli, S.: An overview of automated scoring of essays. *J. Technol. Learn. Assess.* **5**(1), 5–21 (2006)
3. Williamson, D.M.: A framework for implementing automated scoring. In: Annual Meeting of the American Educational Research Association and the National Council on Measurement in Education, San Diego, CA (2009)
4. Chen, H., He, B., Luo, T., Li, B.: A ranked-based learning approach to automated essay scoring. In: 2012 Second International Conference on Cloud and Green Computing (CGC), pp. 448–455 (2012)
5. Yongwei, Y., Buckendahl, C.W., Juszkievicz, P.J., et al.: A review of strategies for validating computer-automated scoring. *Appl. Measur. Educ.* **15**(4), 391–412 (2002)
6. Dumais, S.T.: Latent semantic analysis. *Annu. Rev. Inf. Sci. Technol.* **38**(1), 188–230 (2004)

7. Foltz, P.W., Laham, D., Landauer, T.K.: Automated essay scoring: applications to educational technology. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications, vol. 1999(1), pp. 939–944 (1999)
8. Mesnil, G., Mikolov, T., Ranzato, M., Bengio, Y.: Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. CoRR, abs/1412.5335 (2006)
9. Socher, R., Lin, C.C., Ng, A.Y., Manning, C.: Parsing natural scenes and natural language with recursive neural networks. In: ICML, pp. 129–136 (2011)
10. Zanzotto, F.M., Korkontzelos, I., Fallucchi, F., Manandhar, S.: Estimating linear models for compositional distributional semantics. In: COLING, pp. 1263–1271 (2010)
11. Wang, S., Manning, C.D.: Baselines and bigrams: simple, good sentiment and topic classification. In: ACL (2), pp. 90–94 (2012)
12. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: AAAI, pp. 2267–2273 (2015)
13. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. ICML, vol. 32, pp. 1188–1196 (2014)
14. Chen, H., He, B.: Automated essay scoring by maximizing human-machine agreement. In: EMNLP, pp. 1741–1752 (2013)
15. Shermis, M.D., Burstein, J.C.: Automated Essay Scoring: A Cross-Disciplinary Perspective. Routledge, Abingdon (2003)
16. Briscoe, T., Medlock, B., Andersen, Ø.: Automated assessment of ESOL free text examinations. University of Cambridge Computer Laboratory Technical reports, vol. 790 (2010)
17. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
18. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space In: Proceedings of Workshop at ICLR (2013)
19. Shermis, M.D., Burstein, J.: Handbook of Automated Essay Evaluation: Current Applications and New Directions. Routledge, Abingdon (2013)
20. Yannakoudakis, H., Briscoe, T., Medlock, B.: A new dataset and method for automatically grading ESOL texts. In: ACL, pp. 180–189 (2011)
21. Lawrence, I., Lin, K.: A concordance correlation coefficient to evaluate reproducibility. Biometrics **45**, 255–268 (1989)
22. Croux, C., Dehon, C.: Influence functions of the Spearman and Kendall correlation measures. Stat. Methods Appl. **19**(4), 497–515 (2010)
23. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. Int. J. Forecast. **22**(4), 679–688 (2006)
24. Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. **46**(3), 175–185 (1992)

Knowledge Science, Engineering and Management
10th International Conference, KSEM 2017, Melbourne,
VIC, Australia, August 19-20, 2017, Proceedings
Li, G.; Ge, Y.; Zhang, Z.; Jin, Z.; Blumenstein, M. (Eds.)
2017, XVII, 563 p. 150 illus., Softcover
ISBN: 978-3-319-63557-6