Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

Vector Based Techniques for Short Answer Grading

Ahmed Magooda[1], Mohamed A. Zahran[1], Mohsen Rashwan[2], Hazem Raafat[3] , Magda B. Fayek[1]

[1]Computer Engineering Department, Cairo University, Egypt.

[2]Electronics and Communications Department, Cairo University, Egypt.

[3]Computer Science Department, Kuwait University, Kuwait.

{ahmed.ezzat.gawad, moh.a.zahran}@gmail.com, mrashwan@rdi-eg.com, hazem@cs.ku.edu.kw, magdafayek@ieee.org

Abstract

Vector-based approaches proved their validity during the past few years as promising techniques for word and sentence representation. Automatic short answer grading is a challenging problem in natural language processing that can reduce a lot of human effort, accordingly research was focused towards exploiting several vector representations to solve this problem. In this paper various sentence representation techniques and wide range of similarity measures are compared and finally a system for short answer grading is presented. The system either outperforms the state of the art systems on different data sets or achieves comparable results.

1. Introduction

Assessment is the task of evaluating outcome of an examination process, naturally this task involves human grader. However sometimes the assessment process can represent a huge overhead due to the limited number of graders or the huge number of students which introduces the idea of using Automatic (computerized) assessment.

While some automatic assessment tasks are easy to tackle such as True/False questions and multiple choice questions, there are tasks that represent a challenge to researchers such as Automatic Essay Scoring (AES) and Short Answer Grading (SAG). AES is concerned with scoring essay questions, where answers are likely to be long with no model answer provided. The main trend AES systems follow is to grade the answer by analyzing the spelling, grammar, coherency of sentences (Higgins et al. 2004) and sometimes relation to the main topic. On the other hand SAG is concerned with grading short answers that are about 2~3 sentences of length with the presence of model answer. In SAG the main concern is to grade a student answer in the light of model answer, the grammar and coherency are not of interest in many approaches dealing with SAG.

Unlike some systems which use templates (Pulman et al. 2005) to perform the SAG task, proposed system deals with the SAG problem as a similarity problem following the work of (Mohler et al. 2009, 2011) and the work of (Gomaa et al. 2012, 2014)

Proposed system outputs a grade between 0 and 5, where 0 is totally wrong and 5 is excellent. Bag of words (BOW) representation was used to represent sentences in the proposed system. While calculating similarity the system used two previously proposed representations for calculating sentence similarity in addition to a newly introduced sentence representation (Min Max representation).

The rest of the paper consists of related work (section2), followed by introduction to used data sets (section 3), introduction to similarity measures (section 4), proposed system (section 5) and finally obtained results (section 6).

2. Related Work

Various systems have been proposed to solve SAG, some systems use hand crafted patterns to detect the answered parts, other systems use patterns aided with some machine learning techniques and other systems measure the similarity between the student answer and the model answer.

C-Rater (Leacock et al. 2003) is a system developed by ETS for the task of short answer grading, the system compares syntactic features extracted from the student answer with a set of concepts extracted from model answers.

Oxford-UCLES (Sukkarieh et al. 2004) developed by University of Oxford uses manually crafted patterns to make decision if a specific part was answered correctly, using some words and synonyms the model can be trained to extended manually inferred patterns. In its new implementation the system compares some machine learning methods like decision tree learning, and Bayesian learning.

Indus Marker utilizes question answer markup language to represent the student answer as a structure, after some linguistic analysis the system matches the structure of the student answer with the structures of correct answers to measure the degree of similarity (Siddiqi et al. 2010).

Like Texas system developed by (Mohler et al. 2009, 2011) and the work of (Gomaa et al. 2012), proposed system treats the task of Short Answer Grading as a text similarity task as the two tasks are strongly related, both tasks are based on calculating the similarity between two sentences based on the features extracted from sentences. In

SAG similarity between model answer and student answer is calculated, finally a grade relative to the similarity calculated is assigned to student answer. M.Mohler, focused on combining and comparing corpus based similarity measures as Explicit Semantic Analysis (ESA) (Gabrilovich et al. 2007) and Latent Semantic Analysis (LSA) (Deerwester et al. 1990) with knowledge based measures, he also compared the effect of corpus size and corpus generality and proposed to use graph aligning instead of using BOW model. Gomaa,W.H., combined three types of similarity measures (Corpus based, string based and knowledge based) on Texas data set.

3. Data Sets

Proposed system was evaluated on four data sets.

Texas computer science data set (Mohler et al. 2009) this data set consists of answers submitted for three assignments in the class of computer science. Each assignment consisted of seven short-answer questions. Thirty students were enrolled in the class and submitted answers to these assignments. Thus, the data set consists of a total of 630 student answers. The answers were independently graded by two human graders, using an integer scale from 0 (completely incorrect) to 5 (perfect answer).

Extended Texas computer science data set (Mohler et al. 2011) this data set consists of ten assignments between four and seven questions each and two exams with ten questions each. The data set consists of 81 question, 20 answer for each question were used which sums to 1620 student answer. The data set was graded by two human graders, average of the two human grades was used as final score.

Cairo University data set (Gomaa et al. 2014) this data set consists of 61 question 10 student answers for each. The data were taken from the course of environmental science from Cairo University. The data consists of two versions one is the Arabic (the original data) and the other is English human translation version.

SemEval 2013 data set (Dzikovska et al. 2013) the Joint Student Response Analysis and 8th Recognizing Textual Entailment Challenge at SemEval 2013. Proposed system was evaluated (without participation) on the 5-way task using the SciENTSBank data set. The SciENTSBank data contains training data and 3 types of test data (Unseen Answers (UA), Unseen Questions (UQ) and Unseen Domain (UD)). The following table contains a summary of data size.

	Training	UA	UQ	UD
Model Answers	135	135	15	46
Student answers	36	4	36	65

Table 1. *Summary of SemEval 2013 SciENTSBank data set*

4. Similarity Measures

Similarity measure is a measure to express how much two words are similar either semantically or structurally using real number between 0 ~ 1 where 0 means totally irrelevant and 1 means totally similar. Similarity measures can be classified into three major types.

4.1. String Similarity

String similarity compares two streams of chars and determines the similarity score based on string matching of the two strings regardless of their meaning. One of the famous algorithms for string similarity is Levenshtien Distance which calculates the similarity based on the minimum number of operations that can convert one string to the other, the three operations introduced are addition, removal and substitution. Other string similarity measures are Hamming distance, Damerau-Levenshtein distance, Needleman-Wunsch distance, Smith-Waterman distance, Jaro Winkler distance, Dice's coefficient, Jaccard similarity and longest common substring.

4.2. Knowledge-based Similarity

Knowledge-based similarity is based on using source of knowledge to calculate the similarity between two words. The similarity calculated in this case is semantic similarity that reflects how two words are related not by the structure of words but by the semantic properties of the two words.

Knowledge based similarity are calculated using WordNet (Princeton University 2010) the most used source. WordNet is a large lexical database of English nouns, verbs, adjectives and adverbs that are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. Some of the knowledge based similarity measures are Res, Lin, JiangConrath, Lch, Wup, Shortest path, Hso and Lesk.

4.3. Corpus based Similarity

Corpus based similarity uses the statistical information gained from processing over big corpus to construct some knowledge space that can be used afterwards to calculate relation between words and documents. Calculating semantic similarity using corpus based can be much more tempting due to its statistical nature that does not require pre-built knowledge source which can need a lot of human effort and sometimes cannot be available for all languages.

Some of corpus based similarity measures are LSA, ESA and Extracting Distributionally Similar words using Co-occurrence (DISCO) (Kolb et al. 2008).

DISCO[1] measure, the DISCO similarity measure is based on scanning the corpus by a window of variable width (preferred ±3) then a co-occurrence matrix is constructed using the unique words as rows. Col-

[1] DISCO tool and preprocessed data for many languages can be found on http://www.linguatools.de/disco/disco_en.html

umns are the unique combination of (word, position) pair in the window. The Matrix is then filled by the frequency of co-occurrence between row words and column words.

The absolute values in the matrix are then converted into feature values by applying equation (Eq-1)

$$log \frac{(f(w,r,w') - 0{,}95) f(*,r,*)}{f(w,r,*) f(*,r,w')} \qquad (1)$$

Where w and w' stand for words and r for a window position, and f is the frequency of occurrence. Then the vector comparison between two words is carried out by Lin's information theoretic measure (Eq-2)

$$lin = \frac{\Sigma_{(r,w')}(w_m, {*}r, {*}w') + (w_n, {*}r, {*}w')}{\Sigma_{(r,w')}(w_m, *, {*}w) + \Sigma_{(r,w')}(w_n, *, *)} \qquad (2)$$

The absolute values in the matrix are then converted into feature values by applying equation.

4.4. Word Vector Representation

Word vector representation is the process of representing a word with a vector in high dimensional space, each dimension of the generated space holds semantic or syntactic feature for words. This high dimensionality representation of words is utilized to measure the semantic similarity between words using any distance measure like cosine distance, Euclidian distance and Manhattan distance. Some of the recent word vector representations that achieved high accuracy in many tasks are (Word2Vec (Mikolov et al. 2013) and Global Vectors for Word Representation (GloVe) (Pennington et al. 2014)). Two word vector representations were used in the proposed system and results are reported. Follows a brief description of the two representations.

Word2Vec,[2] Word2Vec is vector representation for words toolkit developed by (Mikolov et al. 2013). The model proposed has a structure similar to Neural Networks while using log linear classifiers as the core of the model. The parameters of the trained log linear classifiers are used as vector representation for words (word embeddings). Two models to train the log linear classifiers were proposed in (Mikolov et al. 2013); Continues bag of words Model and Skip Gram Model. The first trained on predicting word given context and the other is trained on predicting context given word. The similarity between words can be calculated by cosine distance measure (Equation 3) between the two vectors representing words in concern.

The ***GloVe***[3] algorithm is another word vector representation, GloVe utilizes the idea of word co-occurrence relation. GloVe learns word vectors (word embeddings) by building co-occurrence matrix for large corpus, these word vectors are trained to capture global features that are encoded in the ratio between co-occurrence probabilities of words. Cosine distance is used.

$$similarity = cos(\Theta) = \frac{A.B}{||A||\,||B||} = \frac{\Sigma_{i=1}^{n} A_i \text{ x } B_i}{\sqrt{\Sigma_{i=1}^{n}(A_i)^2} \text{ x } \sqrt{\Sigma_{i=1}^{n}(B_i)^2}} \qquad (3)$$

5. Proposed System

Proposed system merges and compares between different similarity measures. Proposed System, consists of three modules. The data preprocessing module, Similarity Measure module and the scaling module.

5.1. Pre-Processing

Different pre-processing techniques were applied including basic text cleaning tasks like (Stop words removal (SWR), stemming and lemmatization[4]). The reported results will show that the preprocessing stage affects the overall performance of the system.

5.2. Similarity measure

The similarity measure module is responsible for calculating a similarity value between 0 ~ 1 for model answer and student answer which indicates how much the two answers are similar. Although the similarity measures reported in this work performs on the scale of word-to-word, the extension to sentence-to-sentence similarity is handled by many methods that will be discussed in sentence similarity section.

5.2.1. Word-to-word similarity

The algorithms used for calculating word-to-word similarity are presented in this section. The extension to sentence-to-sentence similarity will be illustrated later. In this work ***seven*** similarity measures and vector representations were used as features fed to a classifier. One of the measures is string based, two are knowledge based, one is corpus based and three are based on vector representations for words. Follows the measures and representations used.

String Based:

Block distance similarity measure is also known as Manhattan distance and city block distance. It computes the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Block distance between two items is the sum of the differences of their corresponding components (Krause et al. 2012).

Knowledge Based:

JiangConrath Jiang & Conrath calculate the similarity between Word1 and Word2 by the following equation.

$$Sim = \frac{1}{IC(Word1) + IC(Word2) - 2 \text{ x } IC(LCS)} \qquad (4)$$

LCS is least common subsumer
IC (word) = −log P (word)

[2] https://code.google.com/p/word2vec/

[3] http://nlp.stanford.edu/projects/glove/

[4] Stanford core NLP framework was used for stemming and lemmatization " http://nlp.stanford.edu/software/corenlp.shtml "

P (word) is the probability of the word occurrence in large corpora

Lesk Given Word1 & Word2 the Lesk measure is based on counting the shared terms that can be found in the WordNet definition of Word1 & definition of Word2. This measure is based on the sense disambiguation algorithm proposed by Lesk.

Corpus Based:

DISCO, DISCO algorithm was used with the Wikipedia pre-processed corpus provided on the Project website. Preprocessed corpus is 2008 Wikipedia version with corpus size of 267 million words and 220,000 unique words.

Vector Representations:

Word2Vec, the Word2Vec vector representation toolkit was used with the set of vectors released on the project website, the vector set are pre-trained on part of Google News data set (about 100 billion words). The vector set contains 300-dimensional vectors for 3 million words. For Arabic the used vector set contains 300-dimensional vectors for around 6 million words5. Words that are not present in the vector set are represented with out of vocab (OOV) vector which is a tiny value across all dimensions in our case we unified all OOV vectors to the value 0.0001 for all dimensions.

GloVe, the GloVe vector representation algorithm was used with the set of vectors released on the project website, the vector set are pre-trained by Common Crawl (840 Billion tokens). Vector set consists of 300-dimensional vectors for 2.2 million words. For Arabic the used vector set contains 300-dimensional vectors for around 6 million words[6]. Words that are not present in the vector set are represented with out of vocab (OOV) vector.

Sense Aware Vectors[7] (Neelakantan et al. 2014), the sense aware vectors are built on the pre-trained Word2Vec vectors by giving each word multiple vectorized representations, one for each sense. A word sense is represented by its context, which means that similar contexts represent the same sense.

When using the sense vectors first search for the word in the set of vectors, if found count its senses. If just one sense is available use it as a vector, otherwise measure the cosine distance between context vector (sum of vectors of surrounding words) and all the centroids. Sense vector that corresponds to the centroid that maximizes cosine distance is selected. If word not found in sense vectors retrieve the vector from Word2Vec vector set as illustrated in algorithm *Convert word to vector (sense-aware).*

Algorithm: Convert word to vector(sense-aware)

```
Input Word, context, senseWordVectorDictionary, word2VecDictionary
If senseWordVectorDictionary contains Word
    senseVectors = senseWordVectorDictionary[Word]
    If numOfSenses = 1:
        Return sense from senseVectors
    Else:
        contextVector = getContextVector(context)
    For sense in senseVectors:
        cosineDistance = Cosine(senses.Centroid,context)
    Return sense.Vector where sense maximizes cosineDistance
Else:
    Return word2VecDictionary[Word]
```

Vector set released on the website by the authors was used, the set consists of 300-dimensional vectors for 100 thousand words. For Arabic vector representation; the used vector set consists of 300-dimensional vectors for around 200 thousand words trained on Arabic Gigaword corpus and classical Arabic corpus.

5.2.2. Word-to-word similarity

To calculate the similarity between two sentences three models were utilized.

Text-to-text model proposed in (Mihalcea et al. 2006) by associating each word in one sentence to the word that maximizes similarity in the other sentence. Where similarity measure is any similarity measure used. For two Sentences S1&S2 the text-to-text similarity can be calculated using (Equation 5) where ***w*** is word and ***f(w)*** is the frequency of word ***w*** in sentence

$$Similarity(S_1, S_2) = \frac{1}{2} \text{ x } \left(\frac{\sum_{w \in S_1} (MaxSimilarity(w,S_2) \text{ x } f(w))}{\sum_{w \in S_1} f(w)} + \frac{\sum_{w \in S_2} (MaxSimilarity(w,S_2) \text{ x } f(w))}{\sum_{w \in S_2} f(w)} \right) \quad (5)$$

When calculating similarity using **Text-to-text model** two approaches were investigated, either normalizing the final score relative to the length of the two sentences or dealing with it as it is. This model was used to calculate both string based, knowledge based similarity and DISCO from the corpus based similarity.

Vector summation model, for each type of vector representation Word2Vec, GloVe and Sense aware vectors, a sentence is represented as a vector by adding up the vectorized representation of its words. For each type of vector representation cosine similarity is then calculated between the vectors representing the two sentences, these values will be the features used in the next module. This property "Additive Compositionality" is one of the highlighted properties of Word2Vec vector representation. Additive property was utilized in calculating similarity using Word2Vec, Glove and Sense Aware Vectors.

The proposed system also used another variant of the vector summation, which is weighted IDF summation (Zahran et al. 2015). The weighted IDF summation simply

[5] https://sites.google.com/site/mohazahran/data

[6] https://sites.google.com/site/mohazahran/data

[7] https://people.cs.umass.edu/~arvind/emnlp2014wordvectors/

multiply each vector with its IDF value that is extracted from a big corpus and then finally normalization over the IDF summation is performed. This variant takes the word importance in its account instead of weighting all the words with the same weight.

Min-Max Additive model, this is the third model. In this model for each sentence three vectors are combined to represent the final sentence vector. The three vectors are the words summation vector, the maximum vector and the minimum vector.

$$\forall_{i=1}^{n} \mathrm{MaxVector}_{i} = \forall_{w \in s} max(w_{i}) \qquad (6)$$

$$\forall_{i=1}^{n} \mathrm{MinVector}_{i} = \forall_{w \in s} min(w_{i}) \qquad (7)$$

Where $\boldsymbol{W}$ is word vector, $\boldsymbol{W_i}$ is the ith dimension in vector $\boldsymbol{W}$, $\boldsymbol{n}$ is the length of the vectors and $\boldsymbol{S}$ is the set of words that construct the sentence.

The sentence vector is the concatenation of the previously mentioned three vectors. Similarity is then calculated by applying cosine measure to the two vectors.

The idea behind using the Min-Max vectors is trying to represent the sentence with a vector that captures the boundaries of the sentence so that, when calculating cosine similarity it can be interpreted as to what extent the two sentences cross boundaries. This model is applied in calculating similarity using Word2Vec, Glove and Sense Aware Vectors.

5.3. Scaling Module

As the task in concern is a grading task, the output must be an understandable grade that occurs in a well-defined interval of grades. The previously mentioned module "Similarity Module" outputs a value between 0 ~ 1 that indicates how much the two sentences are similar, the task of the scaling module is to map this value to a grade.

In this module **support vector regression (SVR)** (Smola et al. 2004) is used, the features that are fed to SVR are the values generated from the previously mentioned similarity measures while output is the student grade between 0 and 5.

Following (Mohler et al. 2009) and (Gomaa et al. 2012) 10 fold validation is performed over all the data. For SemEval data set provided training set was used as training and validation while testing was performed on the 3 provided test sets (Unseen Question (UQ), Unseen Answers (UA) and Unseen Domains (UD)).

6. Results

The following are the results reported for the four data sets. In the following results system developed by combining Block distance measure, JiangConrath, Lesk, DISCO, Word2Vec, GloVe, and Sense Aware Vectors will be referred to as Vectorized System, on the other hand the system without vector representations which combines Block distance measure, JiangConrath, Lesk and DISCO will be referred to as Basic System.

For the following data sets the data is scored by two human judges so error will is reported relative to average of the two scores. The **IAA** (Inter Annotator Agreement) is calculated between the two scores provided by judges. The reported correlation is Pearson's correlation. For the Cairo University Arabic data, (Gomaa et al. 2014) reported the results of his work on the English translated data and subset of his work on Arabic data.

	Correlation
(Mohler et al. 2009) IAA	0.644
(Mohler et al. 2009) Text-to-Text	0.509
Vectorized – SVR	**0.59**

Table 2. *Results of different models on Texas data set*

	RMSE	Correlation
(Mohler et al. 2011) IAA	0.659	0.586
(Mohler et al. 2011) SVR	0.998	0.518
(Mohler et al. 2011) SVM	0.978	0.464
(Gomaa et al. 2012)	NA	0.504
Vectorized – SVR	**0.91**	**0.550**

Table 3. *Results of different models on Extended Texas data set*

	RMSE	Correlation
(Gomaa et al. 2014) IAA	0.69	0.86
(Zahran et al. 2015)**Arabic**	0.95	0.82
(Gomaa et al. 2014)**English**	0.75	0.83
(Gomaa et al. 2014)**Arabic**	1.07	0.73
Basic System - **Arabic**	1.1	0.78
Vectorized - Arabic	**0.89**	**0.84**

Table 4 *Results Achieved on Cairo University Arabic data set*

	weighted-averageF1			macro-average F1		
System	UA	UQ	UD	UA	UQ	UD
CoMeT1	0.59	0.29	0.25	0.55	0.20	0.15
EHUALM2	0.52	0.44	0.43	0.44	0.35	0.34
ETS2	**0.62**	0.35	0.43	**0.58**	0.27	0.33
SoftCardinality	0.53	0.49	**0.47**	0.47	0.38	**0.37**
UKP-BIU1	0.59	0.39	0.40	0.56	0.32	0.34
Vectorized Approach	0.47	**0.51**	**0.46**	0.37	**0.52**	**0.37**

Table 5. *Results Achieved on SemEval data set*

As we can see in Table 2 from the reported results on Texas data set the proposed vector-based system achieved 0.59 correlation compared to best reported system which achieved 0.51. Although (Mohler et al. 2009) reported correlation only, it can be noticed that vector models are far more superior as it enhanced the results by 8% over reported system.

On the extended Texas data set Table 3 proposed system achieved 0.55 correlation compared to best reported system which achieved 0.518. Proposed system achieved around 4% lower than the Inter Annotator Agreement.

On Cairo University data set Table 4 the proposed system achieved very high correlation and we can see that RMSE achieved is not far behind IAA. We can also notice the enhancement that vector representation models achieved to the basic system as it achieved about 5% reduction in RMSE and 6% correlation enhancement.

In Table 5 We can see the results obtained on SemEval data set. Proposed system achieved the highest result on the unseen questions (UQ) data set with decent margin from all the other systems. For the unseen domain (UD) data set proposed system achieved result with small difference behind the system that came in first place (soft cardinality). On the other hand proposed system achieved not so good results on the unseen answers (UA) data set, it would have achieved the 7th place.

From the results on SemEval data we can infer that the proposed system can generalize very well compared to other systems as the system achieved very good positions in the two generic test sets.

7. Conclusion

In this paper a new state of the art vector-based short answer grading system is proposed. Proposed system deals with the short answer grading problem as text similarity task. A little description of every algorithm or representation used is provided alongside a clarification of the experiment setting to ease reproducing the reported results.

The vector-based proposed system achieved new state of the art results on four data sets with two languages and achieved results that not far behind the Inter annotator agreement. The system proposed combines various types of similarity with main dependency on word vector representation.

For future work we plan to extend the proposed system to cover more Arabic data sets. Some of the research is also directed towards using more sophisticated techniques rather than mere vectors summation such as recursive auto-encoders.

References

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. Journal of the American society for information science 41(6): 391.

Dzikovska, M. O.; Nielsen, R. D.; Brew, C.; Leacock, C.; Giampiccolo, D.; Bentivogli, L.; and Dang, H. T. 2013. SemEval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge, North Texas state University, Denton.

Gabrilovich, E.; and Markovitch, S. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJCAI, 1606-1611.

Gomaa, W. H.; and Fahmy, A. A. 2012. Short answer grading using string similarity and corpus-based similarity. International Journal of Advanced Computer Science and Applications.

Gomaa, W. H.; and Fahmy, A. A. 2014. Automatic scoring for answers to Arabic test questions. Computer Speech and Language 28(4): 833-857.

Higgins, D.; Burstein, J.; Marcu, D.; and Gentile, C. 2004. Evaluating Multiple Aspects of Coherence in Student Essays. HLT-NAACL.

Kolb, P. 2008. Disco: A multilingual database of distributionally similar words. In Proceedings of KONVENS, Berlin.

Krause, E. F. 2012. Taxicab geometry: An adventure in non-Euclidean geometry. Courier Corporation.

Leacock, C.; and Chodorow, M. 2003. C-rater: Automated scoring of short-answer questions. Computers and the Humanities 37(4): 389-405.

Mihalcea, R.; Corley, C.; and Strapparava, C. 2006. Corpus-based and knowledge-based measures of text semantic similarity. AAAI, 775-780.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Effi-cient estimation of word representations in vector space. In proceeding of the International Conference on Learning Representations Workshop track, Arizona, USA.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 3111-3119.

Mikolov, T.; Yih, W. T.; and Zweig, G. 2013. Linguistic Regularities in Continuous Space Word Representations. HLT-NAACL, 746-751.

Mohler, M.; and Mihalcea, R. 2009. Text-to-text semantic similarity for automatic short answer grading. In Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, 567-575.

Mohler, M.; Bunescu, R.; and Mihalcea, R. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 752-762.: Association for Computational Linguistics.

Neelakantan, A.; Shankar, J.; Passos, A.; and McCallum, A. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In Proceedings of EMNLP.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. EMNLP.

Princeton University "About WordNet." WordNet. Princeton University. 2010. http://wordnet.princeton.edu.

Pulman, S. G.; and Sukkarieh, J. Z. 2005. Automatic short answer marking. In Proceedings of the second workshop on Building Educational Applications Using NLP, 9-16.: Association for Computational Linguistics.

Siddiqi, R.; and Harrison, C. J. 2010. Improving teaching and learning through automated short-answer marking. Learning Technologies 3(3): 237-249.

Smola, A. J.; and Schölkopf, B. 2004. A tutorial on support vector regression. Statistics and computing 14(3): 199-222.

Sukkarieh, J. Z.; Pulman, S. G.; and Raikes, N. 2004. Automarking 2: An update on the UCLES-Oxford University research into using computational linguistics to score short, free text responses. International Association of Educational Assessment, Philadephia.

Zahran, M. A.; and Tawfik, A. Y. 2015. Adaptive Tuning for Statistical Machine Translation (AdapT). Computational Linguistics and Intelligent Text Processing, 557-569.

Zahran, M. A.; Magooda, A.; Mahgoub, A. Y.; Raafat, H.; Rashwan, M.; and Atyia, A. 2015. Word Representations in Vector Space and their Applications for Arabic. Computational Linguistics and Intelligent Text Processing, 430-443.