International Journal of Cognitive Informatics and Natural Intelligence
Volume 17 • Issue 1

Efficacy of Deep Neural Embeddings-Based Semantic Similarity in Automatic Essay Evaluation

Manik Hendre, Ramanbyte Pvt. Ltd., India*
Prasenjit Mukherjee, Ramanbyte Pvt. Ltd., India
Raman Preet, Ramanbyte Pvt. Ltd., India
Manish Godse, Pune Institute of Business Management, India

ABSTRACT

Semantic similarity is used extensively for understanding the context and meaning of the text data. In this paper, use of the semantic similarity in an automatic essay evaluation system is proposed. Different text embedding methods are used to compute the semantic similarity. Recent neural embedding methods including Google sentence encoder (GSE), embeddings for language models (ELMo), and global vectors (GloVe) are employed for computing the semantic similarity. Traditional methods of textual data representation such as TF-IDF and Jaccard index are also used in finding the semantic similarity. Experimental analysis of an intra-class and inter-class semantic similarity score distributions shows that the GSE outperforms other methods by accurately distinguishing essays from the same or different set/topic. Semantic similarity calculated using the GSE method is further used for finding the correlation with human rated essay scores, which shows high correlation with the human-rated scores on various essay traits.

KEYWORDS

ELMo, Embedding, Essay Grading, Global Vectors, Semantic Similarity, Sentence Encoder

INTRODUCTION

Automatic Essay Evaluation is one of the oldest research area in the field of Natural Language Processing (NLP). Unlike multiple choice questions and short question answers, an essay is an open ended question. There is no fixed format and one can have multiple ways of writing an essay. Manually grading the essays is a very resource intensive task from the perspective of time and labour. Teachers have to spend their valuable time on grading the essays written by the students. If we have an automatic essay grading system then teachers can devote more time on the teaching part. An essay is used to assess one's understanding of the particular language. Because of which, TOEFL (2019) and GRE (2019) like exams has essay writing as one of the main component. Since last 5 decades researchers are developing solutions for automatic essay grading systems (Page, 1968; Christie, 1999;

DOI: 10.4018/IJCINI.323190 *Corresponding Author

Rudner et al., 2006). In Natural Language Processing field there has been many advancements in last couple of years. We have more powerful language models which can perform various tasks as par with humans (Young et al., 2018). In tasks like sentiment Analysis, Chatbot, Question Answering, Automatic Essay Evaluation, Dialogue Systems, Parsing, Word-sense disambiguation, Named-Entity Recognition, POS Tagging and many more, we are observing good results (Young et al., 2018; Khurana et al., 2017; Cambria & White, 2014). The computing resources are more available and affordable now, as compared with couple of years back. Due to this, the research in NLP using Deep Learning Techniques is taking new leap in every field (Otter et al., 2020; Young et al., 2018; Deng & Liu, 2018).

In this paper, different neural embeddings are used to check their efficacy in automatically evaluating the essays by considering the semantic similarity. Survey of different word embedding methods have been performed by Wang (2019). Specifically, six word embedding models have been evaluated on different natural language processing tasks. In this, authors points out that currently there are no metrics available for evaluation of word embedding models. The semantic and syntactic relations captured by word embedding models are different from how human beings, understand languages (Wang et al., 2019). Most of the Word embedding models are task specific because they are trained for specific natural language processing tasks. In many NLP tasks, we need to compare different set of texts. For natural language understanding, only keyword matching while similarity checking is not sufficient. The semantics have very important role to play in natural language generation and understanding. There are many ways in which same text can be written having the same meaning. The semantics tries to capture this meaning from different text data. In this paper, we are going to calculate semantic similarity using different neural embedding techniques on an essay data. Automatic Essay Evaluation using Word-Mover Distance is proposed by Tashu & Horvath (2018). In This Semantic similarity of text is given more weightage than the syntax and vocabulary. For calculating essay score, the word-mover distance between Normalized Continuous Bag-of-word features is calculated (Wang et al., 2019). Semantic similarity based on knowledge graphs is proposed in (Zhu & Iglesias, 2016). Most of the semantic similarity techniques uses only surrounding words while computing semantic similarity. Knowledge graphs represent concepts and complex relationships can be extracted from them (Zhu & Iglesias, 2016). Semantic similarity in academic articles where length of the document is more based on word embeddings, is proposed in (Liu et al., 2017). To improve the accuracy, authors have proposed to create semantic profile for each article which then will be used along with word embeddings to calculate similarity. Semantic similarity between two words, sentences and paragraphs is presented by Pawar & Mago (2019). In this, sentence similarity is computed in two phases, first phase the similarity is maximized using word, sentence and word-order similarity. In second phase, the skewness is removed which was introduced because of deviation from actual similarity. Automatic evaluation of text using word and sentence embeddings is proposed by Clark et al. (2019). Authors have introduced a new metric sentence mover's similarity which is the extension of word mover distance for multiple sentences. Sentence mover's similarity metric has improved correlation with the human judgment scores on automatic text evaluation task (Clark et al., 2019). In semantic similarity context of a word is important. Context Representation method using bi-direction LSTM is proposed in (Melamud et al., 2016). Few of the recent and notable contribution in the field of an automatic essay evaluation are reviewed in Table 1.

The main contribution of this paper is to calculate neural embeddings based semantic similarity score to be used in an automatic essay evaluation. The rest of the paper is organized as follows. In Section 2, we list all the studied neural embedding techniques. Datasets used are explained in the Section 3. Proposed methodology and the performance evaluation techniques are explained in Section 4. Experimental results are presented in Section 5. Finally, the conclusions are drawn in Section 6.

NEURAL EMBEDDING TECHNIQUES

For every NLP task the numerical vector representation of text data is very important. Most of the machine learning and deep learning techniques require numeric vectors as an input to the system.

Table 1. Existing automatic essay evaluation systems

Sl. No.	Paper	Details
1	Essay Grading System Based on LSA with LVQ and Word Similarity (Ratna et al., 2018)	Word similarity is included into an existing LSA and LVQ based Essay grading system. Word similarity is calculated by counting the number of reference keywords present in an input essay.
2	Essay Scoring using Reinforcement Learning (Wang et al., 2018)	Reinforcement learning based is proposed to train the essay scoring model. Quadratic weighted Kappa metric is used as the reward function. QWK is computed for the pack of essays and grading a single essay is considered as the action taken in the reinforcement learning framework.
3	Automated essay scoring with string kernels and word embeddings (Cozma et al., 2018)	Character level n-gram features which are called as string kernels are combined with the word embeddings for an essay scoring purpose.
4	Automatic Essay Scoring of Swedish Essays using Neural Networks (Lilja, 2018)	Automatic Essay scoring for Swedish using LSTM is proposed.
5	Essay scoring system using N-GRAM (Fauzi et al., 2017)	To take into consideration the word order in an essay grading, N-gram based approach is used.
6	Automatic Features for Essay Scoring An Empirical Study (Dong & Zhang, 2016)	Rather than using the hand crafted features, two-layered convolutional Neural Network (CNN) is used for automatic feature extraction.
7	Automated Essay Grading Based on LVQ and NLP Techniques (Shehab et al., 2016)	Artificial neural network based technique, learning vector quantization is used for training the essay grading model. Additionally, different NLP techniques are used for giving feedback to the students.
8	Automated essay scoring with e-rater V.2 (Attali & Burstein, 2006)	Advanced version of the E-rater is presented with additional features. This version gives more judgmental control in many modelling parameters. Grammatical, organizational, lexical and vocabulary based features are considered in an essay grading.
9	Essay Grading with Probabilistic Latent Semantic Analysis (Kakkonen et al., 2005)	Automatic essay scoring for Finnish language is proposed. Assignment specific knowledge is used to train the model. Probabilistic Latent Semantic Analysis technique is used to compute the semantic similarity. Cosine distance between probability vectors is used as a similarity metric.
10	Automatic Essay Grading Using Text Categorization Techniques (Larkey, 1998)	Bayesian classifier is used to classify essay into good and worse essay. Essay specific 11 features along with the Bayesian and K-nearest neighbor classifier scores are combined using linear regression to predict an essay score.

Traditional method of representing text into vector form is TF-IDF. Term frequency is the number of times a particular word appears in a document. Inverse document frequency assigns a weight to a word according to how rare or common that word is in set of documents. It gives more weightage to rarely occurring words. Product of Term frequency and inverse document frequency is the single number representation of the word in a document. Jaccard Index is a similarity metric widely used for computing similarity of text. Jaccard index calculates similarity as the intersection over union of the words in two set of texts. According to Jaccard index, if there are many words which are common in two set of texts then those texts are more similar. TF-IDF and the Jaccard Index techniques are traditionally used for finding the similarity between text documents. Recently, many artificial neural network based techniques are developed. Widely used neural network based word embedding, Word2vec (Mikolov et al., 2013a, 2013b) is developed by Mikolov et al. at Google. In Mikolov et al. (2013a), the authors have proposed two novel architectures for word embeddings. First architecture is continues Bag-of-words Model which predicts the current word given a context or surrounding words. Second architecture is a continuous skip-gram model which tries to predict context given an input word. The architecture used by authors is shallow network which is less computationally intensive. Several improvements over Mikolov et al. (2013a) are proposed in Mikolov et al. (2013b). As stop-words don't provide much

information regarding semantics the authors have performed sub-sampling of stop words to improve the training speed. Simple mathematical operations like addition and subtraction can be performed on word vectors, which surprisingly gives interesting semantic relationships among words (Mikolov et al., 2013b). The neural embeddings given by Word2Vec are good at maintaining semantic and syntactic structure among words (Mikolov et al., 2013a, 2013b). Sentence level embeddings are proposed in Cer et al. (2018). In this, Universal sentence encoder takes input sentence of any length and gives its 512 dimensional numeric representation. Fixed length representation has advantage over variable length representation in downstream NLP Tasks. Two different approaches for sentence encodings are presented in Universal Sentence Encoder. First approach uses Transformer networks which gives accurate results at the expense of more computational resources. The second approach makes use of Deep Averaging Network which are less accurate as compared with transformer based model but are efficient in terms of speed and memory (Cer et al., 2018). Earlier transfer learning based neural embeddings were word-based but the solution provided in Universal Sentence Encoder is sentence based and these models can be directly used with the help of transfer learning (Cer et al., 2018). Embeddings for Language Model (ELMo) is deep learning based embedding technique proposed by Peters (2018). The Embeddings are computed by using bi-directional language models. Specifically, Long Short Term Memory (LSTM) with forward and backward passes have been employed for training purpose. ELMO is a feature based approach. Unlike other methods in which neural embedding is a function of top layer, In ELMo the final vector representation is the function of all the internal layers. ELMo has shown improvements on large number of NLP tasks (Peters et al., 2018). Another word to vector representation GloVe (Global Vectors) which takes into account global information is developed at Stanford (Pennington et al., 2014). Unlike word2vec which only considers surrounding words while calculating an embedding, the GloVe takes into account the global context. In GloVe, words are projected in a space such that semantically similar words will be adjacent to each other. For global context the word-word co-occurrence statistics are calculated. This method performs well on word analogy task (Pennington et al., 2014).

DATA MANAGEMENT

The dataset Automated Student Assessment Prize (ASAP) (The Hewlett Foundation, 2019) provided under Kaggle competition namely 'The Hewlett Foundation: Automated Essay Scoring' is used for analysis. For this competition, total 12978 essays are collected. These essays are written by students from grade 7 to 10. Essay length is not constant, each essay has typically 150 to 550 words into it. There are eight sets of essays. Out of eight essay sets, essay set 1, 2, 7 and 8 are of persuasive or narrative in nature. Whereas essay sets 3, 4, 5 and 6 are source dependent in which the source text is provided, by studying it student has to write the essays. This dataset has good amount of variation in terms of text data. Each essay has been double scored with the help of human graders. Some of the essays are graded by multiple human graders on different traits. Three types of scores for each essay of the dataset is available consisting of rate1's domain score, rater2's domain score and the resolved domain score among all the raters. To rank the different deep neural embedding techniques, this paper computes the intra-class and the inter-class semantic similarity between the same and different essay sets respectively. The ASAP (The Hewlett Foundation, 2019) dataset has eight different essay sets containing different essay content which allows us to calculate the semantic similarity between the same set's essay as well as the different set's essay. One drawback of the ASAP (The Hewlett Foundation, 2019) dataset is that it contains only the overall scores for 6 of the 8 essay sets. Only two essay sets are evaluated on different essay traits. To overcome this drawback, Mathias and Bhattacharyya (2018) have done the work of annotating the essays on different essay traits. Well qualified human graders were employed for evaluating the essays. The details about the ASAP++ dataset are given in Table 2. Persuasive or argumentative essays are evaluated on the Convention, Organization, Sentence Fluency and Word Choice traits. The source dependent essays are evaluated based on the Content, Prompt Adherence, Language and Narrativity parameters. The original dataset have anonymized the words

like person names, addresses or the words which mentions the personal information. These words are substituted by the personally unidentifiable words like Person1, Person2, and Organization1 etc. This paper makes use of the ASAP++ (Mathias & Bhattacharyya, 2018) dataset to find the correlation among different essay trait's scores and the semantic similarity scores.

METHODOLOGY

In this paper, the use of semantic similarity in an essay scoring system is proposed. In any text based evaluation system, the scoring should be done on the basis of the context or meaning of the text rather than just text matching. Human graders also takes meaning of the written text into the consideration while grading the essays. So if there is a model essay written by the human expert adhering to all the required conditions then one can simply compare this essay with the student written essays. There is no single or fixed way of writing an essay, each student has its own way of writing an essay. That's why we cannot perform the string matching of model essay and the student written essay. This work proposes to calculate the semantic similarity between the model essay and the student written essay. Figure 1 shows the process of using deep neural embedding based semantic similarity in an automatic essay scoring system. The context aware numerical representation of an input essay and the model essay is calculated using different neural embeddings techniques. Similarity between these embeddings is calculated using the Cosine similarity metric. This similarity score can be used to give actual grade to an essay. Higher semantic similarity with the model essay, means the high score for an essay.

Figure 2 explains the process of calculating intra-class and inter-class semantic similarity scores. For Intra-class similarity the input text should be from same set/topic. For calculating Inter-class similarity we have compared text from different set/topics. To calculate the semantic similarity, the text data has to be converted into its numerical vector representation. We call this vector representation as embedding. Let p_{ij} be the embedding of i^{th} essay from j^{th} set. We have in total 8 sets containing essays on eight different topics. Let q_j be the embedding for the model essay from j^{th} set. We use the cosine

Table 2. ASAP++ dataset details

Essay Set	Essay Type	Traits	Score
Set 1	Persuasive or Argumentative	Content, Convention, Organization, Sentence Fluency and Word Choice	1 – 6
Set 2			1 – 6
Set 3	Source Dependent	Content, Prompt Adherence, Language and Narrativity	0 – 3
Set 4			0 – 3
Set 5			0 – 4
Set 6			0 – 4

Figure 1. Neural embedding-based automatic essay scoring process

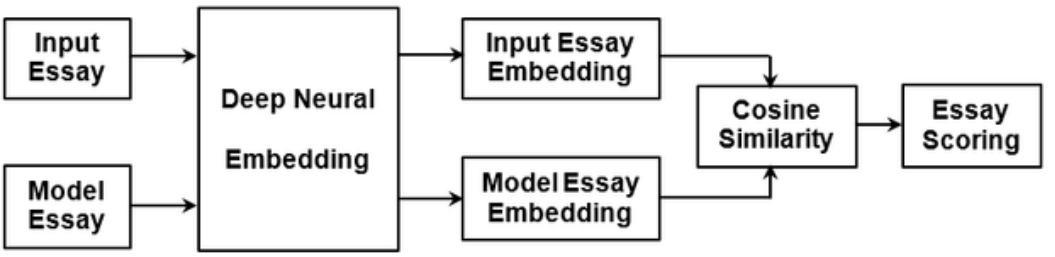
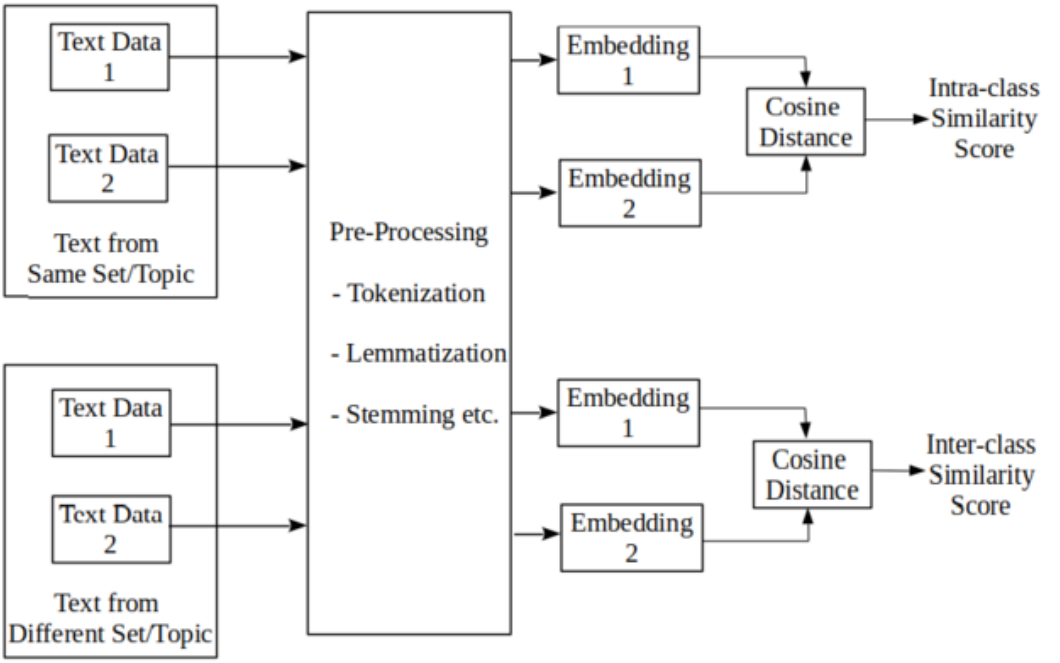


Figure 2. Intra-class and inter-class semantic similarity computation process



similarity metric to calculate the similarity between two set of texts. Cosine similarity computes the angle between the vectors representing the embeddings of two sets of text data.

The cosine similarity is calculated as given in equation (1):

$$Cos\left(\theta\right) = Similarity\left(p_{ij}, q_j\right) = \frac{p_{ij} \cdot q_j}{\left|p_{ij}\right|\left|q_j\right|} \tag{1}$$

When the angle between two embeddings is 0 then we get $Cos\left(\theta\right)$ value as 1 which denotes that the embeddings are exactly similar to each other.

Selection of Model Essay

The datasets used in this paper does not provide the reference or model essay due to which we have selected the top scored essay as the model essay. There can be many essays having the top score, because of this, following steps are taken to find the model essay for each essay set.

Steps to Select Model Essay

Step 1: Find an Essay having maximum domain1 score.
Step 2: If there is only one essay having maximum domain1 score then go to Step 7.
Step 3: Else find the maximum Average All Traits Score for all the essays found in Step 1.
Step 4: If there is only one essay having maximum Average All Traits Score then go to Step 7.
Step 5: Else find the length of each essay found in the Step 3.
Step 6: Return first essay having maximum length as model essay.
Step 7: Return an essay as model essay.

Performance Evaluation

Following performance evaluation criterions are used for comparative analysis of the used neural embedding methods:

1. **Distribution Plot:** In this, we plot the similarity score distribution between intra-class similarity scores and inter-class similarity Scores. Intra-class scores are the ones which are calculated by comparing text from the same essay set. Inter-class scores are the ones which are calculated by comparing text from different essay sets. In the distribution plot, we want maximum separation between curves of intra-class and inter-class scores. More the separation, more accurate the similarity computation method is.
2. **Box Plot:** Box plot shows the five number summary of the similarity scores for each Essay set. For each essay set, we plot both the intra-class and inter-class similarity scores in the same graph. Ideally there should not be any overlap between box plots of the intra-class and inter-class box plots.
3. **D-Prime:** It computes the separation between given intra-class and inter- class probability distributions. D-prime is calculated as given by the equation (2):

$$d' = \frac{\sqrt{2}\left|\mu_{IntraClass} - \mu_{InterClass}\right|}{\sqrt{\sigma^2_{IntraClass} + \sigma^2_{InterClass}}} \tag{2}$$

where, μ is the mean and σ^2 is the variance of similarity score distributions. Higher value of D-Prime shows the better performance.

4. **Correlation With Human-Rated Scores:** In this we have calculated the correlation between the similarity score and the actual grades given by the domain experts. Pearson correlation coefficient is used for computing the correlation. Pearson correlation coefficient(r) is calculated as in equation (3):

$$r = \frac{n\left(\sum xy\right) - \left(\sum x\right)\left(\sum y\right)}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right]\left[n\sum y^2 - \left(\sum y\right)^2\right]}} \tag{3}$$

EXPERIMENTAL RESULTS

In the experimental analysis, first all the methods are compared to check how they perform at semantic similarity task. Top Scored Essay from each essay set is considered as the model essay for comparison purpose. For Intra-class similarity calculation, model essay from each essay set is compared with all the essays from that set only. For Inter-class similarity calculation, the model essay from each set is compared with first 100 essays each, from other essay sets. So we have 12977 intra-class and 5600 inter-class similarity scores for each evaluated method. To compute the distance between neural embeddings, we have used the cosine distance similarity metric. We have used TF-IDF, Jaccard, Glove (Pennington et al., 2014), Google Sentence Encoder Large (Cer et al., 2018), Google Sentence Encoder Lite (Cer et al., 2018) and ELMo (Peters et al., 2018) methods to compute similarity between essay texts. Best performing model is selected for further correlation analysis with that of the human rated essay scores.

Similarity Score Distribution

In this section the Intra-class and Inter-class semantic similarity distributions are shown for each evaluated method. We have also used the Box plots to show how each essay set distributions are performing for all used methods. We have used the same box plot to depict both Intra-class and Inter-class similarity scores. The notched box plots represents the Intra-class semantic similarity scores and the box-plots without notches shows the Inter-class similarity scores. Figure 3 shows the similarity distribution for TF-IDF method. In Figure 3(a), we can see that overlapping region is more. In Figure 3(b), we can see that there is an overlap between the Intra-class and Inter-class similarity scores for essay sets 3 and 7. For all other essay sets, we have good amount of separation. Similarity score distributions for Jaccard Index method are shown in the Figure 4. In Figure 4(b), we can see that, there is an overlap in an essay set 2 and 3.

Figure 5 shows the distribution plots for the GloVe embeddings method. In Figure 5(b), we can see that most of the notched and normal box-plots are overlapping. Which denotes that the underlying score distributions are not significantly different. GloVe method fails in capturing the semantic similarity on essay text data as compared with other methods. We can see the performance of ELMo technique in Figure 6. From Figure 6(b), we can see that, except for essay set 4, all other box plots for intra-class and inter-class distances are well separated. This shows good performance on the semantic similarity task.

Performance of Google Sentence encoder Lite and Large techniques are shown in the Figure 7 and Figure 8 respectively. Both the sentence encoder methods perform well on semantic similarity task. We can see the Distribution plots in which overlap between Inter-class and Intra-class score distributions is less. Also the Box-Plots for maximum essay sets shows the clear separation between Inter and Intra class similarity scores. GSE Large takes more memory and time to compute the embeddings as compared with GSE Lite. But improvement in performance is not that significant. Results of both the GSE-Large and GSE-Lite are almost similar.

D-Prime

D-prime quantifies the separation between two probability distributions. Table 3 shows the d-prime values for all the methods used for evaluation. By observing distribution and box plots we could not distinguish between the performance of GSE-Large and GSE-Lite. But by observing the d-prime values we can see that GSE-Large performs best as compared with other methods including GSE-

Figure 3. TF-IDF semantic similarity distribution

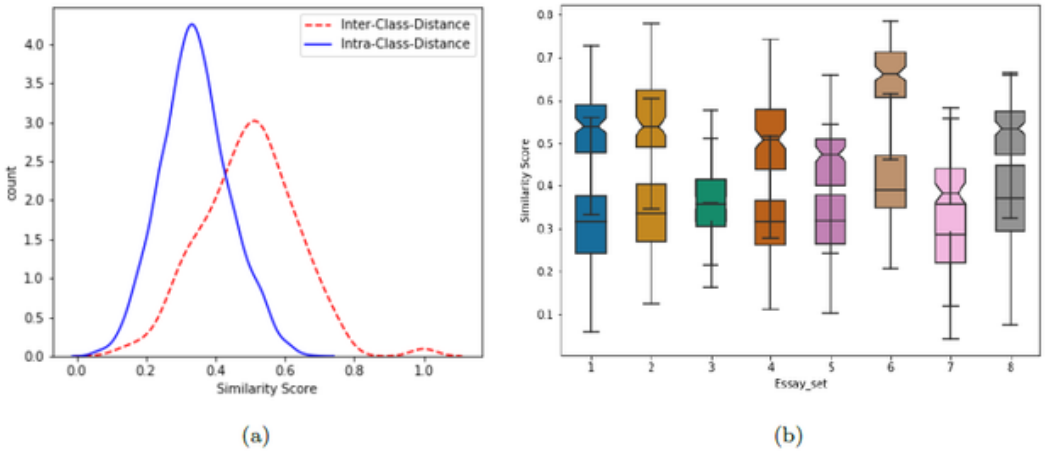


Figure 4. Jaccard semantic similarity distribution

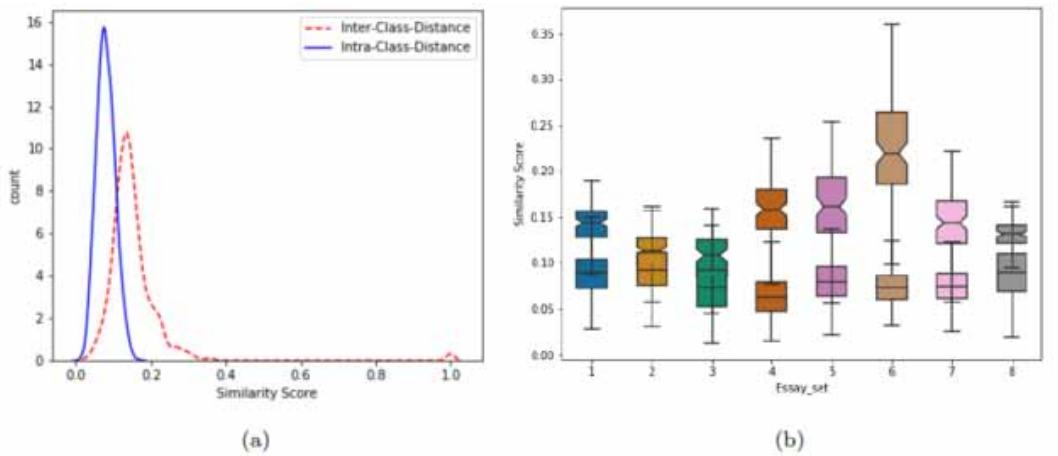
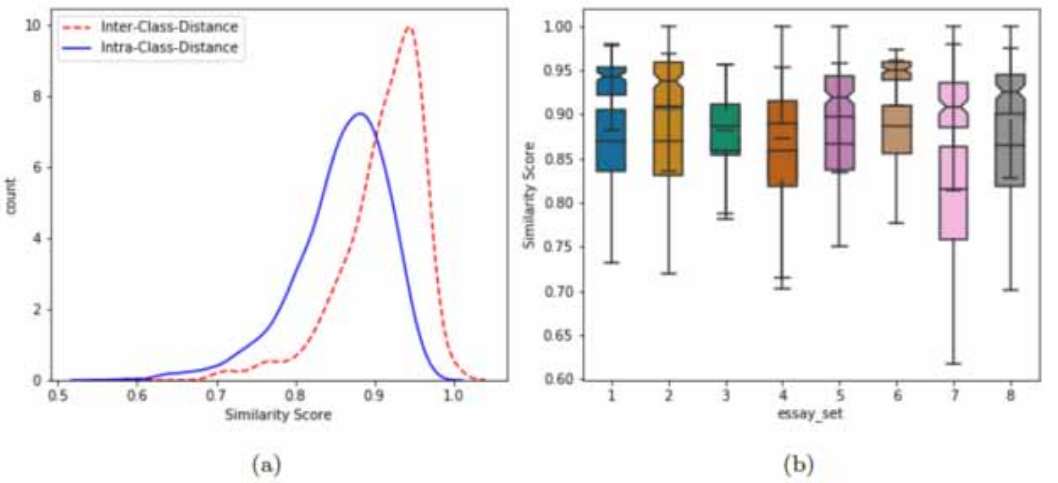


Figure 5. GloVe semantic similarity distribution



Lite. The traditional methods like TF-IDF and Jaccard index gives better performance than GloVe embedding method. GSE-Large with d-prime value of 2.8375 has the best separation between Intra-class and Inter-class semantic similarity scores. GloVe with d-prime value of 0.9271 has the least separation between similarity scores. Which denotes that it could not distinguish between the essays from the same and different sets.

Correlation of Semantic Similarity With Domain Scores

In this paper, the deep neural embedding of the model essay is compared with the other essays from the same set to find the semantic similarity. This paper, claims that the semantic similarity plays an important role in an automatic grading of the essays. To check how the semantic similarity scores correlate with that of the manually human graded scores, the Pearson correlation coefficient is computed between the semantic similarity scores and the human grades. The experimental analysis in this article shows that the Google Sentence Encoder Large (Cer et al., 2018) outperforms all the other

Figure 6. ELMo semantic similarity distribution

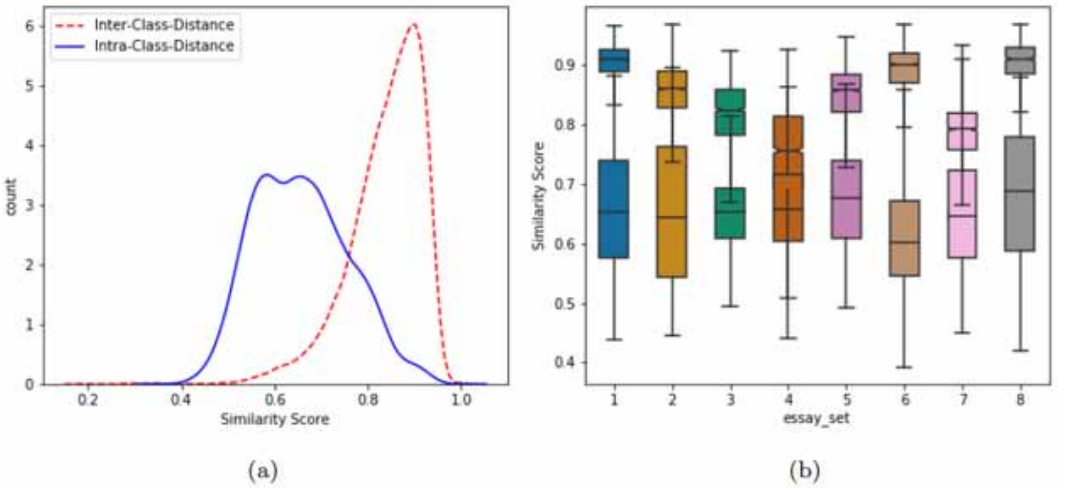
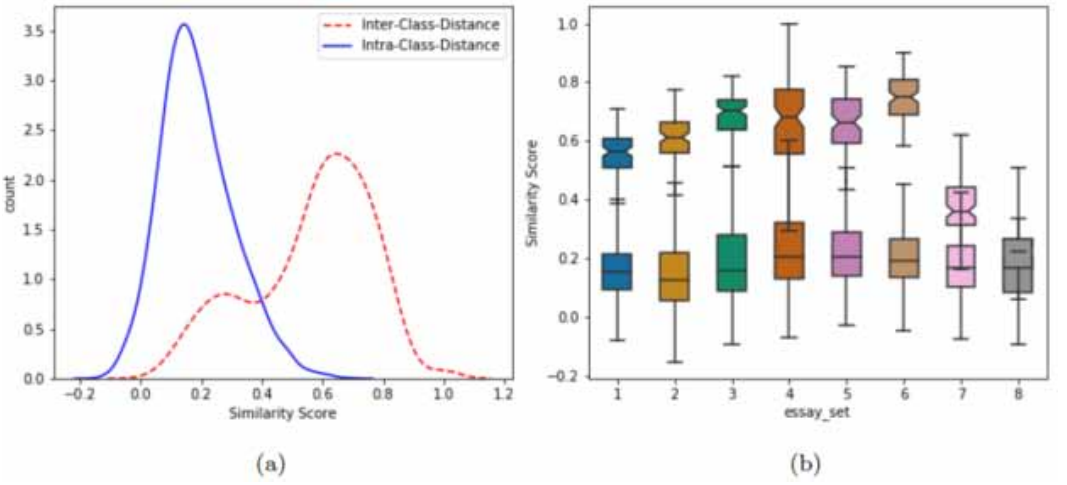


Figure 7. GSE-lite semantic similarity distribution



methods under consideration. Due to this, semantic similarity scores computed using the GSE-Large (Cer et al., 2018) model are used in this section for correlation analysis. The correlation is computed with overall domain scores and essay specific trait's scores. ASAP (The Hewlett Foundation, 2019) has three different types of human graded scores namely domain1 score, rater1 domain1 and the rater2 domain1 scores. Two different raters are used to evaluate each essay and their individual scores are given in rater1 domain1 and the rater2 domain1 scores respectively. The overall scores are provided in the domain1 score. Table 4 shows the correlation between the semantic similarity scores and the domain1 score, rater1 domain1 and the rater2 domain1 scores. The Pearson correlation coefficient of more than 0.5 is considered as the moderate correlation and the value greater than 0.7 is generally considered as a high correlation. One can see from Table (4) that, all the correlation values are greater than 0.5. Essay set1 has the highest correlation of 0.7463 between the semantic similarity scores and the overall domain1 score. Set2 essays has the highest correlation with similarity scores given by rater1 domain1 and the rater2 domain1 scores as compared with other essay sets.

Figure 8. GSE-large semantic similarity distribution

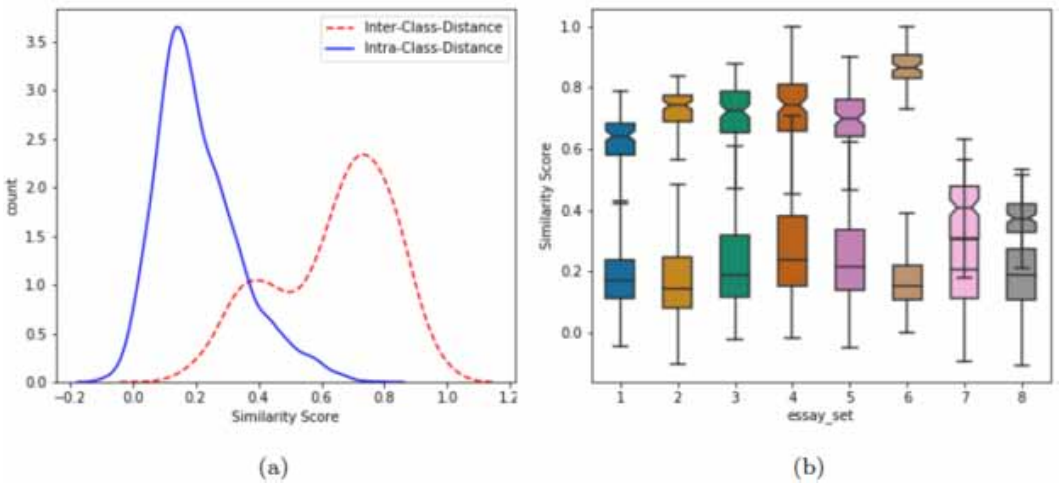


Table 3. Intra-class and inter-class separation

Method	D-Prime
GSE Large	2.8375
ELMo	2.1527
Jaccard	1.6013
TF-IDF	1.2434
GSE Lite	1.2349
GloVe	0.9271

ASAP++ dataset (Mathias & Bhattacharyya, 2018) has provided the scores for the 6 sets of essays according to specific essay traits. This dataset has human grades, for the first two persuasive or argumentative essay sets on the Content, Convention, Organization, Sentence Fluency and the Word Choice traits. Table 5 shows the correlation of semantic similarity scores with that of the essay specific traits for the persuasive type essays. Set1 has the highest correlation of 0.6910 with that of the Content trait as compared with the other essay traits. Set2 shows the high correlation of 0.6293 with Organization trait as compared with the other essay traits. Table 6 shows the correlation values for the source dependent essays. The source dependent essays are evaluated by the human graders based on the Content, Prompt Adherence, Language and Narrativity traits. All the source dependent essays shows the high correlation with the Content parameter of the essay as compared with the other parameters.

Correlation analysis between semantic similarity and the human rated scores as depicted in Tables 4-6 strongly advocates the use of the deep neural embeddings based semantic similarity in an automatic essay evaluation.

CONCLUSION

In this work, in-depth comparative analysis of the different text embedding methods is performed to check their efficacy in an automatic essay evaluation task. Experimental analysis, shows that

Table 4. Correlation with domain scores

Essay SET	Domain1 Score	Rater1 Domain1	Rater2 Domain1
Set1	0.7463	0.6886	0.6960
Set2	0.6985	0.6985	0.7000
Set3	0.5495	0.5305	0.5204
Set4	0.6576	0.6345	0.6346
Set5	0.7207	0.6962	0.6954
Set6	0.7267	0.6984	0.6999

Table 5. Correlation with specific traits for persuasive essays

Essay Set	Set 1	Set 2
Content	0.6910	0.6240
Convention	0.6206	0.5411
Organization	0.6328	0.6293
Sentence Fluency	0.6281	0.5681
Word Choice	0.6559	0.5892

Table 6. Correlation with specific traits for traits for source dependent essays

Essay Set	Set 3	Set 4	Set 5	Set 6
Content	0.5803	0.6549	0.6406	0.6535
Prompt Adherence	0.5802	0.6636	0.6081	0.6474
Language	0.5330	0.5605	0.5916	0.6147
Narrativity	0.5741	0.6353	0.6188	0.6430

the semantic similarity plays an important role in an automatic grading of essays. Different neural Embedding based techniques are employed for finding the semantic similarity between the essay text data. To calculate the similarity between essay texts, classical methods like TF- IDF and Jaccard Index are used. Advanced deep learning based methods including ELMo, GloVe and Google Sentence Encoder are also employed. Neural embeddings given by the ELMo and Google Sentence encoder gives good results as compared with other methods. GSE-Large with d-prime value of 2.8375 gives the best performance by distinguishing between text from same essay set and different essay sets. Though simple and basic, the TF-IDF and Jaccard index also shows performance comparable to advanced deep learning methods. Extensive correlation analysis is performed by comparing semantic similarity scores with human rated essay scores. Semantic similarity scores computed with the help of GSE-Large shows high correlation with human rated domain scores. The high correlation is also observed in an essay specific traits like Content, Organization, Sentence Fluency, Word Choice, Prompt Adherence, Language and Narrativity. This research offers valuable insights, on which embedding method should be employed, to compute the semantic similarity in an automatic essay evaluation system.

REFERENCES

Attali, Y., & Burstein, J. (2006). Automated essay scoring with e-rater R v. 2. *The Journal of Technology, Learning, and Assessment*, 4.

Cambria, E., & White, B. (2014). Jumping nlp curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, *9*(2), 48–57. doi:10.1109/MCI.2014.2307227

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., Constant, N., Guajardo-espedes, M., Yuan, S., & Tar, C. (2018). *Universal sentence encoder*. arXiv preprint arXiv:1803.11175.

Christie, J. R. (1999). Automated essay marking for both style and content. In *Proceedings of the Third Annual Computer Assisted Assessment Conference*. Loughborough University.

Clark, E., Celikyilmaz, A., & Smith, N. A. (2019). Sentence movers similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 2748-2760). doi:10.18653/v1/P19-1264

Cozma, M., Butnaru, A. M., & Ionescu, R. T. (2018). *Automated essay scoring with string kernels and word embeddings.* 10.18653/v1/P18-2080

Deng, L., & Liu, Y. (2018). *Deep learning in natural language processing*. Springer. doi:10.1007/978-981-10-5209-5

Dong, F., & Zhang, Y. (2016). Automatic features for essay scoring-an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1072-1077). doi:10.18653/v1/D16-1115

Fauzi, M. A., Utomo, D. C., Setiawan, B. D., & Pramukantoro, E. S. (2017). Automatic essay scoring system using n-gram and cosine similarity for gamification based e-learning. In *Proceedings of the International Conference on Advances in Image Processing* (pp. 151-155). doi:10.1145/3133264.3133303

GRE. (2019). *ETS*. https://www.ets.org/gre

Kakkonen, T., Myller, N., Timonen, J., & Sutinen, E. (2005). Automatic essay grading with probabilistic latent semantic analysis. In *Proceedings of the second workshop on Building Educational Applications Using NLP* (pp. 29-36). doi:10.3115/1609829.1609835

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2017). *Natural language processing: State of the art, current trends and challenges.* arXiv preprint arXiv:1708.05148.

Larkey, L. S. (1998). Automatic essay grading using text categorization techniques. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 90-95). doi:10.1145/290941.290965

Lilja, M. (2018). *Automatic essay scoring of Swedish essays using neural networks*. Academic Press.

Liu, M., Lang, B., Gu, Z., & Zeeshan, A. (2017). Measuring similarity of academic articles with semantic profile and joint word embedding. *Tsinghua Science and Technology*, *22*(6), 619–632. doi:10.23919/TST.2017.8195345

Mathias, S., & Bhattacharyya, P. (2018). Asap++: Enriching the asap automated essay grading dataset with essay attribute scores. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* (pp. 51-61). Berlin, Germany: Association for Computational Linguistics. doi:10.18653/v1/K16-1006

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). *Efficient estimation of word representations in vector space*. arXiv preprint arXiv:1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 3111–3119.

Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*. PMID:32324570

Page, E. B. (1968). The use of the computer in analyzing student essays. *International Review of Education*, *14*(2), 210–225. doi:10.1007/BF01419938

Pawar, A., & Mago, V. (2019). Challenging the boundaries of unsupervised learning for semantic similarity. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 16291–16308. doi:10.1109/ACCESS.2019.2891692

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543). Academic Press.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proc. of NAACL.*

Ratna, A. A. P., Arbani, A. A., Ibrahim, I., Ekadiyanto, F. A., Bangun, K. J., & Purnamasari, P. D. (2018). Automatic essay grading system based on latent semantic analysis with learning vector quantization and word similarity enhancement. In *Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality* (pp. 120-126). Academic Press.

Rudner, L. M., Garcia, V., & Welch, C. (2006). An evaluation of intellimetric essay scoring system. *The Journal of Technology, Learning, and Assessment*, 4.

Shehab, A., Elhoseny, M., & Hassanien, A. E. (2016). A hybrid scheme for automated essay grading based on lvq and nlp techniques. In *2016 12th International Computer Engineering Conference (ICENCO)* (pp. 65-70). IEEE.

Tashu, T. M., & Hor_ath, T. (2018). Pair-wise: Automatic essay evaluation using word mover's distance. *CSEDU,* (1), 59-66.

The Hewlett Foundation. (2019). *Automated Essay Scoring*. https://www.kaggle.com/c/asap-aes/

TOEFL. (2019). *ETS*. https://www.ets.org/toefl

Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C.-C. J. (2019). *Evaluating word embedding models: Methods and experimental results.* arXiv preprint arXiv:1901.09785.

Wang, Y., Wei, Z., Zhou, Y., & Huang, X.-J. (2018). Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp.791-797). doi:10.18653/v1/D18-1090

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. doi:10.1109/MCI.2018.2840738

Zhu, G., & Iglesias, C. A. (2016). Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, *29*(1), 72–85. doi:10.1109/TKDE.2016.2610428