International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Dec-2021)

http://dx.doi.org/10.12785/ijcds/1001122

Efficacy of Deep Neural Embeddings based Semantic Similarity in Automatic Essay Evaluation

Manik Hendre[1], Prasenjit Mukherjee[2], Raman Preet[3]and Manish Godse[4]

[1,2,3,4]Artificial Intelligence Group, RamanByte Pvt. Ltd. Pune, Maharashtra, India.

Received 6 Jun. 2021, Revised 2 Aug. 2021, Accepted 13 Nov. 2021, Published 28 Dec 2021

Abstract: Deep neural embeddings are widely used in natural language processing (NLP) applications like question answering, prediction of next word or sentence, translation of language, word sense disambiguation, and many such applications. Recent methods like Google Sentence Encoder (GSE), Embeddings for Language Models (ELMo), and Global Vectors (GloVe) are also engaged in NLP. Traditional methods such as TF-IDF and Jaccard index are also beneficial in NLP. One of the primary steps performed by these methods is to determine semantic similarity, which is at the core of automatic essay evaluation. In this paper, we have proposed to use semantic similarity for an automatic essay evaluation. We have utilized all these text embedding methods to compute semantic similarity on the dataset of essays provided by the Center for Indian Language Technology (CFILT), IIT Bombay. Our experimental analysis of semantic similarity score distributions shows that the GSE outperforms other methods by accurately distinguishing essays from the same or different sets. Semantic similarity calculated using the GSE method is further used for finding the correlation with human-rated essay scores. Correlation of semantic similarity scores with different essay-specific traits given in the ASAP++ dataset is also performed.

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

1. Introduction

Automatic Essay Evaluation is an important research area in Natural Language Processing (NLP) to automate evaluation and scoring. Unlike multiple-choice questions and short question answers, an essay is an open-ended question. There is no fixed format; students can write an essay in multiple ways. Manually grading essays is a resource-intensive task requiring time and labor. Teachers have to spend their valuable time grading the essays of students. If teachers have an automated essay grading system, then it will be helpful for them to speed up the evaluation process. They can devote more time to teaching and mentoring students.

An essay is the core of language learning to develop writing skills. Hence all language assessment exams have essay writing as one of the main components, and to mention TOEFL [1] and GRE [2] has mandatory essay. In the last five decades, researchers are developing solutions for automatic essay grading systems [3,4,5]. A good amount of research has been completed in natural language processing to automate many tasks done by humans [6]. Many applications have been implemented and mention few are - language translation, chatbot, sentiment analysis, word-sense disambiguation, and many more. The results of these systems are good [6, 7, 8]. Increased computation power with reduced cost and growing research in deep learning has also supported natural language processing in enhancing results [6, 9, 10].

Wang et. al. [11] in their paper have surveyed several word embedding methods. They have evaluated six models of word embedding against many NLP jobs. They further pointed the unavailability of fair metrics to evaluate word embedding models. They have also mentioned that relations captured by word embedding models for semantic and syntactic are not the same as the way humans perceive languages [11]. In this paper, we are using several neural embeddings to validate their efficacy in automatically evaluating the essays with consideration of semantic similarity.

Word-embedding models can be task-oriented because they may be trained to manage a particular task. Thus they are inefficient in the evaluation of essays. During numerous NLP jobs, a diverse collection of texts may be compared. In this comparison, only keyword matching and their similarity may be insufficient because there may be multiple ways to write the sentence with the same meaning but separate grammar, words, and construction. Hence semantics certainly plays a crucial role in natural language generation and understanding. The semantics attempts to apprehend the meaning from

E-mail:manik.h@ramanbyte.com; prasenjit.m@ramanbyte.com, ramanpreet@ramanbyte.com ,

different writings. This paper also attempts to calculate semantic similarity using diverse neural embedding techniques on essays. This semantics score is useful in deciding the performance of a candidate in an essay.

Tashu et al. [12] proposed word-mover distance to calculate similarity required in automatic essay evaluation. Wang et al. [11] also used the word-mover distance to determine essay score by giving more weight to semantic similarity of text than the syntax and vocabulary. Zhu et al. [13] applied knowledge graphs based semantic similarity in their work. In general, the semantic similarity techniques use surrounding words to determine the semantic similarity, whereas knowledge graphs use the position and the frequency of the concepts in the knowledge graph. Liu et al. [14] suggested a calculation of semantic similarity in academic articles using the length of the document based on word embeddings. Pawar et al. [15] gained accuracy in their work using a combination of a semantic profile of an article and word embeddings to determine the similarity between two words, sentences, and paragraphs. Clark et al. [16] proposed to use the word and sentence embeddings in two stages for the automatic evaluation of a text. The first stage determines similarity by maximizing word, word order and sentence. The second stage is all about removing skewness text. They have suggested a new mover's similarity metric. It is the extension of word mover distance useful when using multiple sentences. Sentence mover's similarity metric has enhanced correlation with the scores of human judgment. Melamud et al. [17] presented the method for context representation using bi-direction LSTM.

In Table (I), we have listed some recent notable contributions in the field of automatic essay evaluation.

TABLE I. EXISTING AUTOMATIC ESSAY EVALUATION SYSTEMS

Sl. No.	Paper	Details
1	Essay Grading System Based on LSA with LVQ and Word Similarity [18]	Word similarity is included in an existing LSA and LVQ based Essay grading system. Word similarity is computed by adding the number of reference keywords present in an input essay.
2	Essay Scoring using Reinforcement Learning [19]	Reinforcement learning 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 framework.
3	Automated essay scoring with string kernels and word embeddings [20]	Character level n-gram features are called as string kernels and they are combined with the word embeddings for an essay scoring.
4	Automatic Essay Scoring of Swedish Essays using Neural Networks [21]	Automatic Essay scoring for Swedish using LSTM is proposed.
5	Essay scoring system using N-GRAM [22]	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 [23]	A two-layered convolutional Neural Network (CNN) is applied in automatic feature extraction instead of the hand crafted features.
7	Automated Essay Grading Based on LVQ and NLP Techniques [24]	Artificial neural network based 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 [25]	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 [26]	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 [27]	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.

In this paper, our main contribution is to calculate the semantic similarity score required in automatic essay evaluation using neural embeddings. We have used different deep neural embedding methods to get the semantic similarity in essay text. Detailed correlation analysis of the semantic similarity scores in comparison with human-rated scores is presented in this paper. The paper has six sections. Section two discusses neural embedding techniques. Section three explains datasets used in experiments. Details on methodology are provided in section four along with the performance evaluation techniques. Experimental results are presented in section five. Finally, the conclusion is discussed in section six.

2. NEURAL EMBEDDING TECHNIQUES

Text data representation in the forms of numerical vectors is an essential requirement in NLP. It is also an input for many machine learning and deep learning methods. Traditionally texts are expressed as TF-IDF vectors based on the word count [35]. TF-IDF highlights words of interest, where TF (Term Frequency) summarizes term frequency in a document and IDF (Inverse Document Frequency) gives a frequency of words across documents. IDF attaches additional weightage to the words which occur not frequently. The result of TF and IDF is the unique number description of the word in a document. Jaccard Index [36] is also a popular similarity metric extensively used by NLP. It calculates the intersection over the union of the words in two sets of texts. Hence depending upon the frequency of common words, it decides the similarity. More the common words more are the similarity.

The TF-IDF and the Jaccard Index methods are common to find the similarity between text documents. Recent advances in the artificial neural network, including deep learning, have inducted new methods like word embedding and Word2vec. The neural embeddings provided by Word2Vec are good for semantic and syntactic structure among words [28,29].

Mikolov et al. at Google [28] have proposed two innovative approaches based on shallow-network for word embeddings. The good part of their work is computationally less intensive shallow-network. The initial approach uses the given context or surrounding words to predict the current word using the bag-of-words technique. The second approach uses a continuous skip-gram to identify the context of a given input word. Later they have also suggested improvement [29] over their earlier work [28]. In order to improve the training speed sub-sampling of stop words is used by them [29].

Cer et al. [30] have proposed sentence-level embeddings. They have used fixed-length representation over the variable length. They have adopted the universal sentence encoder to create a 512-dimensional numeric presentation of input sentences having any sentence size. They have adopted two strategies for sentence encodings-Transformer Networks is the first and the second is Deep Averaging Network.

The transformer network approach gives accurate results but requires more computational resources, whereas Deep Averaging Network provides quick results but is less accurate and consumes fewer resources. The models developed using the universal sentence encoder can be used with the help of transfer learning.

ELMo (Embeddings for Language Model) is a deep contextualized model for complex characteristics of word use. Peters et al. [31] suggested the use of ELMo in their work. A bi-directional language model, which can be pre-trained, is adopted to compute embeddings in ELMo. In model training, LSTM (Long Short Term Memory) is used with forward and backward passes. ELMO is a feature-based approach with the final vector representation as to the function of all the internal layers. It has notable gains in several NLP applications.

GloVe (Global Vectors) is the algorithm developed at Stanford to get vector representations for words. It is a log-bilinear model with a weighted least-squares objective, which considers the global context rather than only surrounding words while calculating an embedding. Hence it is not similar to word2vec. GloVe uses the non-zero global word-word co-occurrence statistics in training the model. This method performs well on word analogy tasks [32].

3. DATA MANAGEMENT

In this paper, Automated Student Assessment Prize (ASAP) [33] essay database is used. This database is publicly available in the Hewlett Foundation: Automated Essay Scoring Kaggle competition. This database has 12978 essays. All the essays are collected from the school students of standard 7 to 10. There was no restriction on the number of words that needs to be used to write an essay. All the essays in the database are having 150 to 550 words length. The essays are written on 8 different topics. Out of these eight sets of essay topics, 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. 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 rater1'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 calculates the semantic similarity among the same topic essays (Intra-Class) and different topic essays (Inter-Class). As the essays in the ASAP [33] database contains eight different sets of essays, it enables the semantic similarity computation between same and different types text.

One drawback of the ASAP [33] 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 [34] 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 the Table (II). 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++ [34] dataset to find the correlation among different essay trait's scores and the semantic similarity scores.

TABLE II. 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

4. METHODOLOGY

We have proposed the use of semantic similarity for automated essay scoring. 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. The 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.

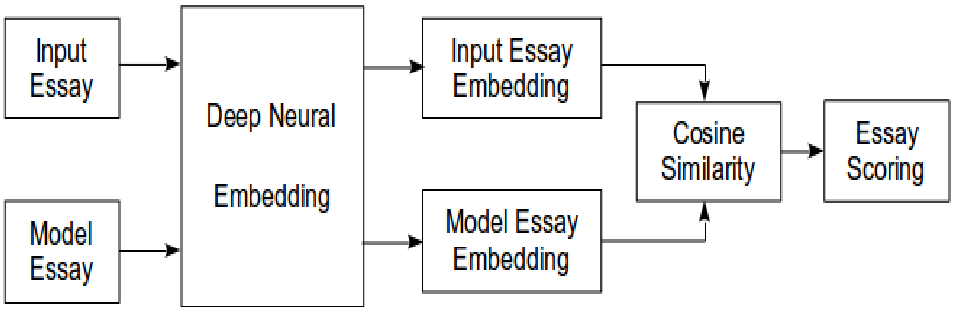


Figure 1. Neural Embedding based Automatic Essay Scoring Process

The figure (2) explains the process of calculating intra and inter class similarity scores. In computing intra-class scores, the embeddings of same topic essays are used. While in computing inter-class scores, the embeddings of different topic essays are used. This way of comparing the text data from same set and different set is inspired from the genuine and impostor comparisons made in biometric recognition. In the recognition field the similarity to the same user data is expected to be high and with other users, it is expected to be low. The input has to be in the numerical form rather than in the text format, while calculating the distance or similarity.

Different embedding techniques can be employed to embed the text data into its numerical form. The numerical representation text data is called as the 'Embedding'. Let's denote m_{xy} as the embedding of x^{th} essay from y^{th} set. The y will take values from 1 to 8 as there are eight essay sets in the database. Let's consider n_y as the embedding for the model essay from particular y^{th} essay set. To calculate the similarity some kind of distance metric is required. In this research paper, the Cosine distance metric is used to compute similarity between embeddings. The cosine angle between two embedding vectors is measured to compute the similarity. The equation (1) shows the formula for cosine similarity calculation.

$$Cos(\theta) = Similarity\left(m_{xy}, n_y\right) = \frac{m_{xy} \cdot n_y}{|m_{xy}|\,|n_y|} \qquad (1)$$

If the angle between two embedding vectors is '0' then Cosine of angle '0' gives value as '1'. The cosine similarity of '1' indicates total similarity between the embeddings.

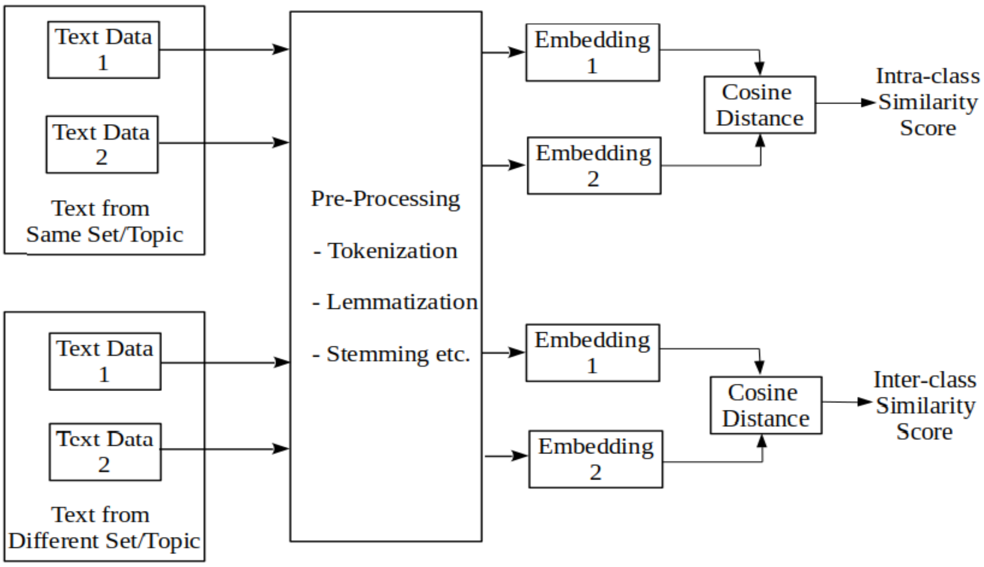


Figure 2. Semantic Similarity Computation Process

4.1 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 goto step 7
Step 3: Else find the maximum AverageAllTraitsScore for all the essays found in step 1
Step 4: If there is only one essay having maximum AverageAllTraitsScore then goto 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 essay as model essay

4.2 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 the same topic essays (intra-class) and different topic essays (inter-class). Intra-class scores are the ones that are calculated by comparing text from the same essay set. Inter-class scores are the ones that 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. The more the separation, the more accurate the similarity computation method is.

2. Box Plot: The box plot shows the five-number summary of the similarity scores for each Essay set. For each essay set, we plot both the similarity scores of same topic essays (intra-class) and different topic essays (inter-class) in the same graph. Ideally, there should not be any overlap between box-plots of semantic similarity scores within intra and inter class.

3. Decidability Index measures the difference between probability distributions. In this paper, the decidability index is used to calculate probability distributions of similarity score and their difference among intra-class and inter-class. Decidability Index is calculated as given in equation (2),

$$d' = \frac{\sqrt{2}\,|\mu_{IntraClass} - \mu_{InterClass}|}{\sqrt{\sigma^2_{IntraClass} + \sigma^2_{InterClass}}} \qquad (2)$$

Here, the mean is μ, and the variance of semantic similarity scores is σ^2. Higher value of Decidability Index 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,

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

5. Experimental Results

In the experimental analysis, the first step is to compare methods for their performance in calculating semantic similarity. For measuring objectives, we have used top-scored essays as a model essay from each set of essays. The intra-class similarity is computed using essays from the same set of essays. The computation of similarity score for all essays is done by comparing them with the model essay provided they belong to the same set.

The score for similarity within inter-class is calculated using essays from the different sets. The model essays are identified from every set of essays. They are the benchmarks for calculating similarity scores by comparing them with the essays from sets. For our experiment, we have used initial 100 essays in scoring. In our experiment, similarity scores are determined using the multiple evaluation methods, and the outcome of the experiment is- 12977 intra-class and 5600 inter-class scores. The distance among neural embeddings is measured using cosine similarity.

The methods employed in the similarity calculations are TF-IDF [35], Jaccard [36], Google Sentence Encoder Large [30], Google Sentence Encoder Lite [30], ELMo [31], and Glove [32] methods. The best-performing model is further selected for correlation analysis with that of the human-rated essay scores.

5.1 Similarity Score Distribution

This section presents distributions of semantic similarity within intra and inter class for each method used in the evaluation. Box plots have been used to present the similarity score of the essay set for every method. The box plots with notches describe the similarity scores of intra-class semantic, whereas box-plot without notches show the similarity scores for inter-class.

Figure (3) shows similarity distribution of TF-IDF. We can see the more overlapping region in figure (3a). For essay set-3 and set-7, the overlap between similarity scores of the intra and inter class can be seen in figure (3b). Rest of the essay sets shows a significant separation. Figure (4) presents distribution of similarity score calculated using Jaccard Index. In essay set-2 and set-3 overlap can be seen in figure (4b). The distribution plots in figure (5) are for the GloVe method. Most of the box plots with and without notch are overlaying and can be seen in figure (5b). Figure (7) shows the outcome of GSE-Lite, while figure (8) displays for GSE-large. Both methods deliver very well toward semantic similarity scores. This can be seen in the distribution plots with less overlap among inter and intra class. Furthermore, the box-plot from most of the sets confirms the definite parting among inter and intra class similarity score. GSE-Lite requires less time and memory to estimate the embeddings as contrasted to GSE-Large. However, there is no improvement in performance. Outcomes for GSE-Large and GSE-Lite are substantially alike.

Figure (7) shows the outcome of GSE-Lite, while figure (8) displays for GSE-large. Both methods deliver very well toward semantic similarity scores. This can be seen in the distribution plots with less overlap among inter and intra class. Furthermore, the box-plot from most of the sets confirms the definite parting among inter and intra class similarity score. GSE-Lite requires less time and memory to estimate the embeddings as contrasted to GSE-Large. However, there is no improvement in performance. Outcomes for GSE-Large and GSE-Lite are substantially alike.

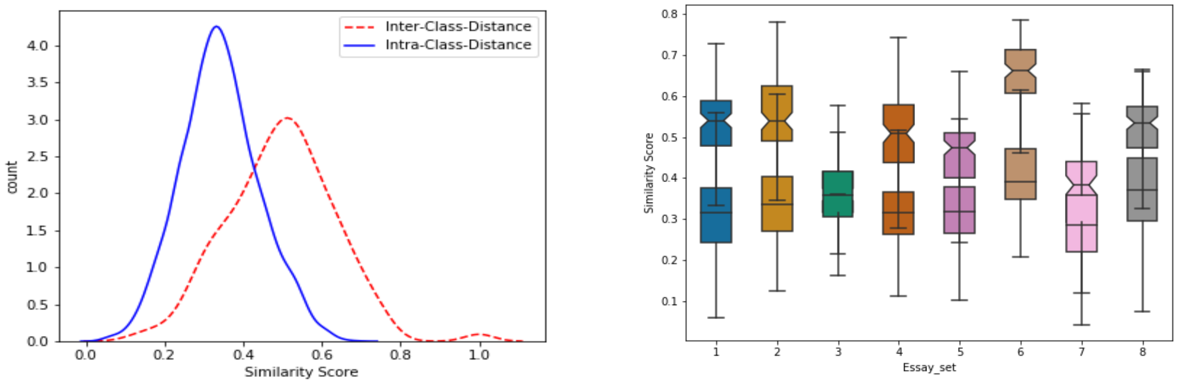
5.2 Decidability Index

Table-III presents values of the decidability index for all the methods used in the evaluation. Decidability Index represents the separation between two probability distributions. The distribution and box plots are not able to differentiate the performance of GSE-Large and GSE-Lite. However, decidability index values indicate that GSE-Large performs most reliable than remaining methods inclusive of GSE-Lite.

The TF-IDF and Jaccard index have performed well compared to the GloVe method. With 2.8375 as decidability index, GSE-Large is best in separating similarity scores between intra-class and inter-class semantics. GloVe method scores at the bottom with a decidability index value as 0.9271. It means GloVe is unable to differentiate the essays from the same and/or different sets.

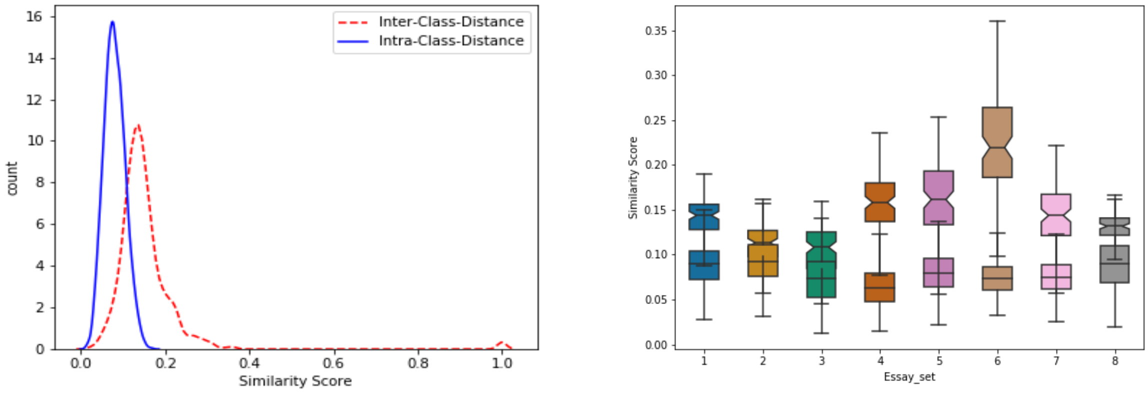
Table III. Separation In Intra and Inter class

Method	Decidability Index
GSE Large [30]	2.8375
ELMo [31]	2.1527
Jaccard [36]	1.6013
TF-IDF [35]	1.2434
GSE Lite [30]	1.2349
GloVe [32]	0.9271



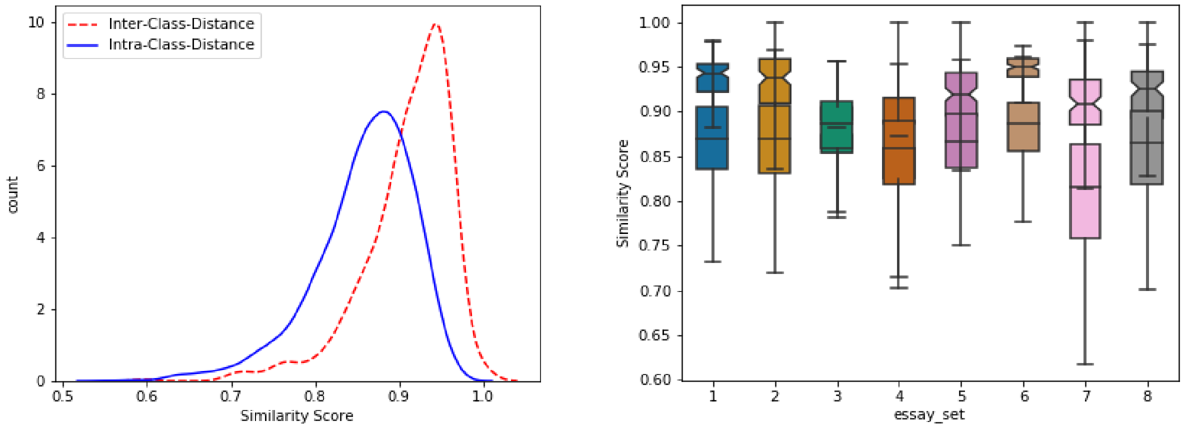
(a) (b)

Figure 3. TF-IDF Semantic Similarity Score Distribution



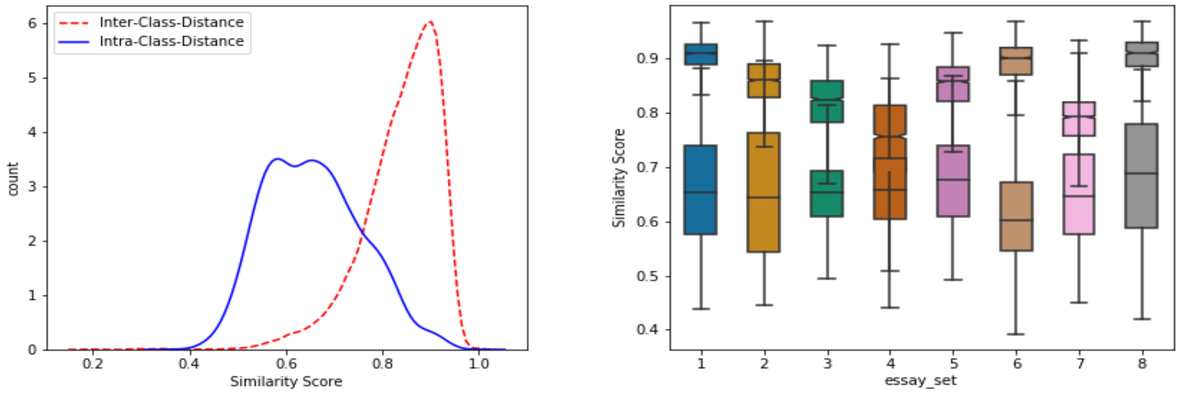
(a) (b)

Figure 4. Jaccard Semantic Similarity Score Distribution



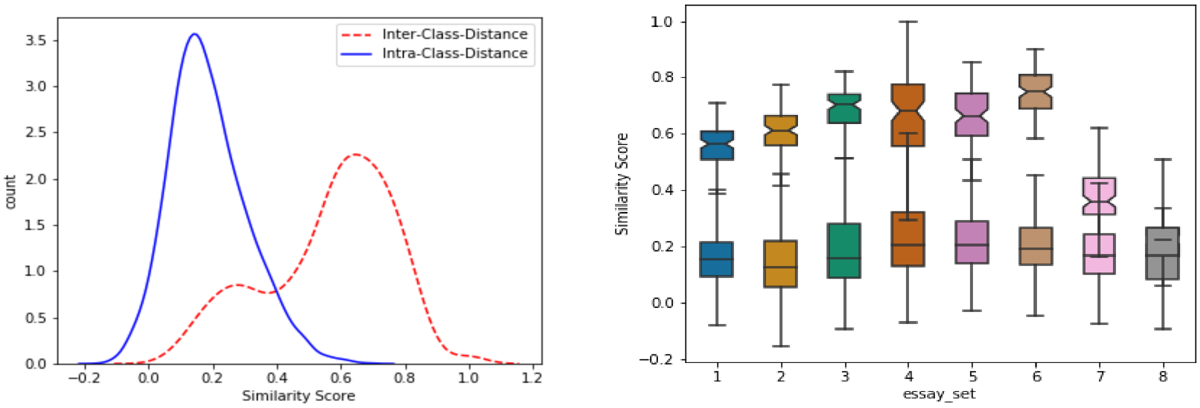
(a) (b)

Figure 5. GloVe Semantic Similarity Score Distribution



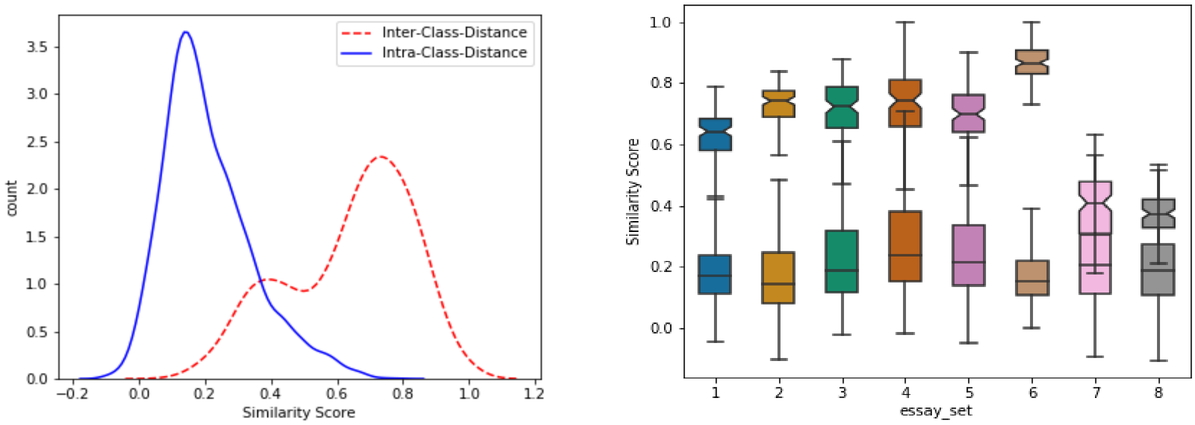
(a) (b)

Figure 6. ELMo Semantic Similarity Score Distribution



(a) (b)

Figure 7. GSE-Lite Semantic Similarity Score Distribution



(a) (b)

Figure 8. GSE-Large Semantic Similarity Score Distribution

5.3 Correlation of Semantic Similarity with Domain Scores

In this paper, the deep neural embedding of the model essay text is equated with other essays from the same set to find the semantic similarity. This paper claims that semantic similarity is crucial in the automatic grading of essays. Pearson correlation coefficient is computed between the semantic similarity scores and the human grades. It is required in examining how the semantic similarity scores correlate with that of the manually human graded scores. The experimental analysis in sections (5.1) and (5.2) shows that the Google Sentence Encoder Large [30] outperforms all the other methods under consideration. Due to this, semantic similarity scores computed using the GSE-Large [30] model are used in this section for correlation analysis. The correlation is computed with overall domain scores and essay-specific trait scores. ASAP [33] has three different types of human graded scores namely domain1 score, rater1 domain1, and rater2 domain1 scores. Two different raters are used to evaluate each essay and their scores are given in rater1 domain1 and rater2 domain1 scores respectively. The overall scores are provided in the domain1 score. Table (IV) shows the correlation between the semantic similarity scores and the domain1 score, rater1 domain1, and 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 (IV) 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 have the highest correlation with similarity scores given by rater1 domain1 and the rater2 domain1 scores as compared with other essay sets.

ASAP++ dataset [34] has provided scores for six 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 (V) shows the scores for correlation of semantic similarity with the characteristics specific 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 (VI) shows the correlation values for the source-dependent essays. Humans are the assessors for source-dependent essays, and grading parameters are content, prompt adherence, language, and narrativity traits. The source-dependent essay shows a high correlation with the content parameter as compared with the other parameters.

TABLE IV. 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 V. 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 VI. CORRELATION WITH SPECIFIC 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

Correlation analysis between semantic similarity and the human rated scores as depicted in the tables (IV, V and VI). It strongly advocates semantic similarity using deep neural embeddings in an automatic essay evaluation.

6. CONCLUSION

In this research paper, an in-depth analysis of the different text embedding methods is performed to check their efficacy in an automatic essay evaluation. Analysis of experiments shows that semantic similarity is one of the key components of automated essay evaluation. The paper has used several recent deep neural embedding techniques for computing the semantic similarity of essay text. The traditional text embedding techniques like Jaccard similarity index & TF-IDF are common in estimating the similarity scores for semantics. The recent embedding schemes based on deep neural are-ELMo, Google Sentence Encoder (GSE-Lite and GSE-Large), and GloVe are also used for computing essay text embeddings.

The research findings of this paper show that the Google Sentence Encoder and ELMo models outperform other embedding methods. The GSE-Large model with 2.8375 Decidability Index gives the highest separation between the semantic similarity scores. The traditional TF-IDF and Jaccard similarity index methods show good performance in determining the semantic similarity of essay text. Extensive correlation analysis is done by comparing semantic similarity scores with evaluation scores given by a human. Semantic similarity scores computed with the help of text embeddings given by GSE-Large show a high correlation with human-rated domain scores. The high correlation is also observed in the 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 automated evaluation of an essay.

REFERENCES

[1] TOEFL: ETS. https://www.ets.org/toe (2019). Online; Accessed: 24 August 2019

[2] GRE: ETS. https://www.ets.org/gre (2019). Online; Accessed: 26 August 2019

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

[4] Page, E.B.: The use of the computer in analyzing student essays. International review of education pp. 210-225 (1968)

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

[6] Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. ieee Computational intelligence magazine 13(3), 55-75 (2018)

[7] Cambria, E., White, B.: Jumping nlp curves: A review of natural language processing research. IEEE Computational intelligence magazine 9(2), 48{57 (2014)

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

[9] Deng, L., Liu, Y.: Deep learning in natural language processing. Springer (2018)

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

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

[12] Tashu, T.M., Horvfiath, T.: Pair-wise: Automatic essay evaluation using word mover's distance. In: CSEDU (1), pp. 59-66 (2018)

[13] Zhu, G., Iglesias, C.A.: Computing semantic similarity of concepts in knowledge graphs. IEEE Transactions on Knowledge and Data Engineering 29(1), 72-85 (2016)

[14] Liu, M., Lang, B., Gu, Z., Zeeshan, A.: Measuring similarity of academic articles with semantic profile and joint word embedding. Tsinghua Science and Technology 22(6), 619-632 (2017)

[15] Pawar, A., Mago, V.: Challenging the boundaries of unsupervised learning for semantic similarity. IEEE Access 7, 16,291{16,308 (2019)

[16] Clark, E., Celikyilmaz, A., Smith, N.A.: 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 (2019)

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

[18] Ratna, A.A.P., Arbani, A.A., Ibrahim, I., Ekadiyanto, F.A., Bangun, K.J., Purnamasari, P.D.: 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 (2018)

[19] Wang, Y., Wei, Z., Zhou, Y., Huang, X.J.: 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 (2018)

[20] Cozma, M., Butnaru, A.M., Ionescu, R.T.: Automated essay scoring with string kernels and word embeddings. arXiv preprint arXiv:1804.07954 (2018)

[21] Lilja, M.: Automatic essay scoring of swedish essays using neural networks (2018)

[22] Fauzi, M.A., Utomo, D.C., Setiawan, B.D., Pramukantoro, E.S.: 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 (2017)

[23] Dong, F., Zhang, Y.: Automatic features for essay scoring-an empirical study. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1072-1077 (2016)

[24] Shehab, A., Elhoseny, M., Hassanien, A.E.: 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 (2016)

[25] Attali, Y., Burstein, J.: Automated essay scoring with e-rater R v. 2. The Journal of Technology, Learning and Assessment 4(3) (2006)

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

[27] Larkey, L.S.: 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 (1998)

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

[29] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111-3119 (2013)

[30] Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)

[31] Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)

[32] Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532{1543 (2014)

[33] Automated Student Assessment Prize: The Hewlett Foundation: Automated Essay Scoring. https://www.kaggle.com/c/asap-aes/ (2019). Online; Accessed: 2 January 2019

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

[35] Teller, Virginia. "Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition." Computational Linguistics 26.4 (2000): 638-641.

[36] Hamers, Lieve. "Similarity measures in scientometric research: The Jaccard index versus Salton's cosine formula." Information Processing and Management 25.3 (1989): 315-18.

Manik Hendre has 6 years of professional experience. He is a Ph.D. research scholar at the Savitribai Phule Pune University. His current role is Data Scientist in RamanByte Pvt Ltd Pune. His research field is - biometrics image processing, NLP, and machine learning.

Prasenjit Mukherjee has 13 years of professional. He holds Ph.D. from National Institute of Technology (NIT), Durgapur, India under the Visvesvaraya PhD Scheme from 2015 to 2019. Presently, He is working as a Data Scientist in RamanByte Pvt. Ltd., Pune. His research is focused on natural language processing and deep learning.

Raman Preet has more than 15 years of professional experience. He has completed his Bachelor of Engineering degree in computer science from University of Pune. He is currently working as Chairman and Executive Director of PIBM group of Institutes. His research areas of interest include Artificial Intelligence, Machine Learning and Education Technology.

Dr. Manish Godse has 30 years of professional. He holds Ph.D. from Indian Institute of Technology, Bombay (IITB). He is currently working as an Industry Professor and IT Director in the PIBM. His research interest is in automation and artificial intelligence.