IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 10, 2019, accepted July 25, 2019, date of publication August 5, 2019, date of current version August 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2933354

Syntactic, Semantic and Sentiment Analysis: The Joint Effect on Automated Essay Evaluation

HARNEET KAUR JANDA, ATISH PAWAR, SHAN DU, AND VIJAY MAGO
DaTALab, Department of Computer Science, Lakehead University, Thunder Bay, ON P7B 5E1, Canada
Corresponding author: Vijay Mago (vmago@lakeheadu.ca)

The work of V. Mago was supported by NSERC Discovery Grant under Grant RGPIN-2017-05377.

ABSTRACT Manual grading of essays by humans is time-consuming and likely to be susceptible to inconsistencies and inaccuracies. In recent years, an abundance of research has been done to automate essay evaluation processes, yet little has been done to take into consideration the syntax, semantic coherence and sentiments of the essay's text together. Our proposed system incorporates not just the rule-based grammar and surface level coherence check but also includes the semantic similarity of the sentences. We propose to use Graph-based relationships within the essay's content and polarity of opinion expressions. Semantic similarity is determined between each statement of the essay to form these Graph-based spatial relationships and novel features are obtained from it. Our algorithm uses 23 salient features with high predictive power, which is less than the current systems while considering every aspect to cover the dimensions that a human grader focuses on. Fewer features help us get rid of the redundancies of the data so that the predictions are based on more representative features and are robust to noisy data. The prediction of the scores is done with neural networks using the data released by the ASAP competition held by Kaggle. The resulting agreement between human grader's score and the system's prediction is measured using Quadratic Weighted Kappa (QWK). Our system produces a QWK of 0.793.

INDEX TERMS Natural language processing, semantic analysis, sentiment analysis and text mining, automated essay evaluation.

I. INTRODUCTION

Essay writing is used in many academic disciplines as a form of evaluation. Generally, a human grader assesses and assigns a score to an essay submission which is written concerning an essay's prompt. This is a laborious and tiring task for the graders. Also, human graders can be imperfect; they are susceptible to biases and irregularities based on other chores and activities they do in life [1]. Different human graders also have different grading styles and can also tend to give an overall higher grade just based on one good impression regarding a particular aspect. A computer system can overcome all these human shortcomings by uniform assessment throughout. Understanding human language is considered a laborious task due to its complexity. There are numerous ways to arrange words in a sentence. Also, words can have multiple meanings in different contexts. Therefore context-based knowledge is necessary to decipher the sentences correctly.

In 1966, Ellis Batten Page presented the idea of an automated system to grade essays based on his own experiences [2] and developed a system called Project Grade Essay® [3]. Although there have been plenty of innovations and advancements in the field since then, most of the existing systems aim to predict scores by taking into consideration extensive number of features which are mostly grammatical flaws, syntax errors and surface level semantic comparison using latent semantic analysis, tf-idf and content importance model [4], [5]. In spite of using a large number of independent prediction variables extracted from the text, most of the systems fall short of in-depth analysis of semantic and sentiment analysis of the essay. In this article, we propose an automated essay scoring (AES) system which uses a fewer number of high-quality, independent variables and provides the essence of the essay which is used to accurately predict the score.

Previous study [1] has shown that when AES is compared with human graders about crucial characteristics of a good essay, the top responses are about the analysis of how the essay revolves around the question prompt, how well structured and sleek the information flow is, quality of grammar used, length, spellings, and punctuation. With respect to these responses, features are extracted from the essays and then

The associate editor coordinating the review of this manuscript and approving it for publication was Huazhu Fu.

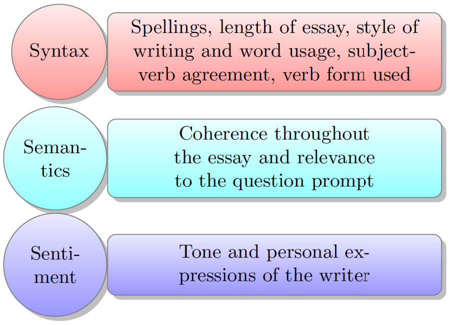

Syntax
Spellings, length of essay, style of writing and word usage, subject-verb agreement, verb form used
Semantics
Coherence throughout the essay and relevance to the question prompt
Sentiment
Tone and personal expressions of the writer

FIGURE 1. **Overview of syntax, semantics and sentiments involved in an essay.**

the most influential ones are selected for an essay grading prediction model. The main strength of our proposed system is to use novel graph based semantic features together with syntactic and sentimental features in order to improve the accuracy of the already proposed systems while minimizing the number of total features used. Different prediction models are tested to find the one which works best for predicting the essay scores.

II. RELATED WORK

The history of automated writing evaluation goes long back. In January 1966, Ellis Batten published an article emphasizing on the possibility and importance of an automated essay evaluation system [2]. Two years later, he successfully developed Project Grade Essay ® [3] which uses statistical methods and multiple linear regression. Intelligent Essay Assessor (IEA) developed by Peter Foltz and Thomas Landauer in 1997 uses natural language processing (NLP), latent semantic analysis (LSA) and machine learning (ML) for the prediction [6]. Latent semantic analysis [7] refers to extracting context-based word meaning from a large corpus by statistical methods. In 1998, a system called Intellimetric® [8] was developed which uses NLP methods and mathematical models for predicting grades. The educational and testing services use e-rater ® [9] for generating scores and feedback which uses NLP to produce features which are combined with a statistical model to predict the scores [9]. Bayesian Essay Test Scoring system developed by Lawrence Rudner uses Bayesian network and statistical methods. The system CRASE® [10] also uses NLP and ML. For efficient processing of text, different areas of natural language processing (NLP) domain, i.e., syntax, semantics, and sentiments are analyzed by the existing systems [1], as shown in Figure 1. Following is a brief description of methodologies used in the literature:

A. SEMANTIC ANALYSIS IN ESSAY EVALUATION

Semantic information on written text tells about coherence and closeness of the information flow within the essay and with the question prompt. There have been attempts to extract semantic information using various NLP techniques in AES. Existing systems measure coherence in the text using different supervised and unsupervised approaches. Usually, the unsupervised approaches measure lexical cohesion, i.e., repetition of words and phrases in an essay. Foltz et al. [11] assume that coherent texts contain a high number of semantically similar words and measure coherence as a function of relatedness between adjacent sentences. Some systems have used latent semantic analysis (LSA) [11], probabilistic latent semantic analysis (PLSA), and latent Dirichlet allocation (LDA) [12]. LSA, PLSA, and LDA are topic modeling techniques, i.e., class of text analysis methods that analyze groups of words together to capture how the meaning of words depends on the broader context in which they are used in natural language. Systems have also used unsupervised approaches like usage of similar words and sentences in an essay to depict the level of coherence [13]. Klebanov *et al.* [14] aimed to predict the score for an essay based on its relatedness to Content Importance models. Higgins *et al.* [4] measured coherence features and incoherence through semantic similarity between essay question and discourse elements of the essay. Zupanc and Bosnić [15] incorporated coherence features to convert parts of essay into a semantic space and measure various characteristics, but the authors failed to provide enough information for other researchers to repeat their results. Semantic relationship between chunks of text can be represented as a graph. The Graph-based features can help obtain information about the coherence in the text by pattern recognition [16]. In natural language, semantic analysis is about understanding the meaning of what is written in a particular text [17]. The semantic part of language processing tries to understand if the formation of the sentences, occurrences of the words in a written/oral communication make any sense or not. Semantic similarities are useful to understand the coherence of the essays and their relevance to the question [15]. Recent works in the NLP area to provide solutions for calculating semantic similarities can be categorized as following methods:

- Word co-occurrence methods [18].
- Similarity based on a lexical database [19].
- Method based on web search engine results [20].
- Methods based on word vectors using recursive neural networks [21].
- Unsupervised approaches [22].

Unsupervised approaches to determine semantic similarities require less computational resources. To dervive graph based features we use unsupervised method used by Pawar *et al.* [22] to get semantic similarity between sentences. This algorithm outperforms other methodologies as per the Rubenstein and Goodenough (R&G) benchmark standard [23]. This algorithm uses an edge-based approach using a lexical database and incorporates corpora-based statistics into a standardized semantic similarity algorithm.

B. SENTIMENT ANALYSIS IN ESSAY EVALUATION

Distinctive opinions and polarity of words used by the writer in an essay shape up the overall essay construction and quality, specifically in persuasive and argumentative essays [24]. Many NLP tasks have used sentiment analysis such as in social media [25], movie's reviews [26], news and politics [27]. One of the first attempts at incorporating sentiments in AES involved using subjective lexicon(s) to get the polarity of the sentences [28]. Some other noteworthy works are finding argumentative discourse in persuasive essays [29] where authors proposed to classify argument components as support or not to obtain argumentative discourse structures. Another prominent work [30] found the sentiment of sentences in essays by examining the sentiment of multi-word expressions. Farra *et al.* [31] matched up to the opinion expressions in the essays to their respective targets and use features extracted to predict the scores.

C. GRAPH-BASED METHODS FOR TEXT ANALYSIS

Understanding written or spoken text is a challenging task, especially for a computer. For information retrieval from the text, there has been an enormous amount of research mainly in the area of text summarization and visualization. Graph-based representation of chunks of texts has been used by many researchers for NLP tasks like text summarization, term disambiguation, and relation extraction [32], [33]. The Graph-based representation helps to discover associations and patterns within a chunk of data. The graph representation of text also allows both the structure and content of documents to be represented [34]. The approach and methods used for what a node or edge represents in a graph can vary based on the type of problem being solved.

Author Pillutla, Venkata Sai Sriram used Graph-based representation to depict the associations between sentences in a text using cosine similarity [35]. Jin et al. used Graph-based representation in text mining task to detect unapparent links between concepts across documents [36]. Graph edges representing order-relationship between the words represented by nodes have been used for text summarization [37]. Giabbanelli et al. used analysis of casual maps to assess problem solving skills of students [38]. Gupta et al. assess student learning in form of causal networks or maps [39]. Giabbanelli et al. use the assessment of knowledge maps to study student's knowledge for an ill structured problem [40].

III. METHODOLOGY

For an efficient natural language processing of a written text, each sub-domain including syntax, semantics, and sentiments should be analyzed. With respect to these sub-domains, we extract features from the essays and most influential ones are selected for an essay grading prediction model. The aim of our proposed system is to use syntactic, semantic and sentimental features together to improve the accuracy of the already proposed systems in the field involving a minimum number of features possible. Different prediction models are then tested to find the one that provides the most accurate predictions for the essay scores. How the development is carried out in different areas of the proposed system has been elaborated in the following subsections.

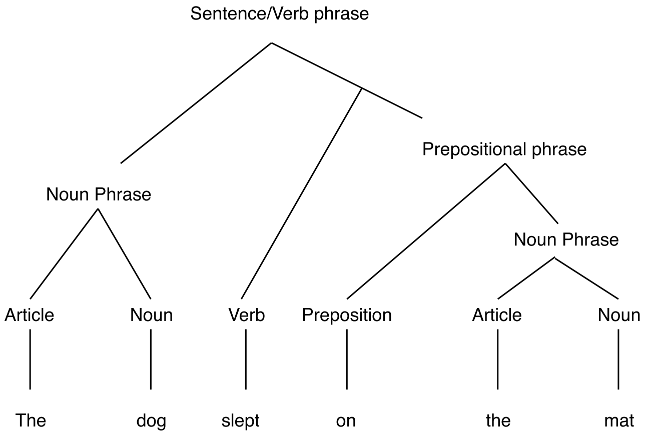


FIGURE 2. Constituent structure of a sentence displaying each part of speech.

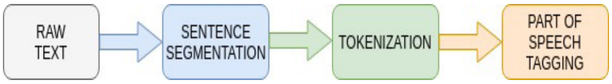


FIGURE 3. Segmentation and tokenization of the essay to determine part of speech used.

A. SYNTACTICAL ATTRIBUTES

Syntax refers to the order/arrangement of content. Word classes, largely corresponding to traditional parts of speech (e.g, noun, verb, preposition, etc.), are syntactic categories. In phrase structure grammars, the phrasal categories (e.g, noun phrase, verb phrase, prepositional phrase, etc.) are also syntactic categories. In this part, we mainly use Natural language processing toolkit (NLTK)[1] [41], which is a python based platform to extract language-based data. NLTK also provides functions like classification, tokenization, stemming, tagging, parsing which are very helpful to extract the syntactic features of a text. To analyze the syntax, a submitted essay is segmented into sentences by given full stops. These sentences are further tokenized into words to analyze each word the essay is composed of, Figure 3. To obtain a set of syntactic features, we selected the ones which are widely used by researchers in the literature [42], [43]. To understand the syntactic structure of an essay, we count the total occurrences of the following syntax related features:

1) **Unique part-of-speech used**: Over repetition of words of the same `part of speech` in an essay is regarded as an inefficient use of English grammar. It is a common mistake made by non-native speakers in their writing [44]. For example, *Tom is a student. Tom is a good guy.* indicates an excessive use of nouns, where a pronoun could have been used instead. Figure 2 shows a sentence after `part of speech` tagging is done. We can see there are four unique `part of speech` used in the sentence. An essay is tokenized into words using *word_tokenize* function from *NLTK* and each word is tagged to a part of speech using the

[1] https://www.nltk.org/

function *pos_tag*. The *found* tags are put into sets. A set is collection with no duplicates. Length of the set is obtained which represents the total number of unique `part of speech` in the essay. We believe using a very limited `part of speech` in an essay can lead to less score.

2) **Misspelled words**: Use of wrong spellings can lead to misinterpretation of the word by the essay evaluator. Also, this is one of the most common mistakes naive writers make [45]. A dictionary for American English called *en-US* is used upon the spell check library pyEnchant to find the number of misspelled words per essay. We use the *check* function from spell checker library and count the total occurrences of misspelled words.
3) **Existential *there***: Existential *there* is used to indicate the presence of existence of an entity [46]. Huckin and Pesante [47] stated that expert writers use the word *there* for only important linguistic purposes like to emphasis existence, to state new information, topics, and to summarize. Therefore, we believe excessive use of existential there can lead to low score and should be avoided. We count the number of existential there using *pos_tag* function from *NLTK* the acronym for a existential there is *EX*. When a `part of speech` tag is found to be *EX*, the counter for existential there is incremented.
4) **Superlative adjectives**: These are used when a subject is compared to three or more objects and usually have a suffix -est added to it. For example, *sweetest* and *brightest*. We count the total occurrences of superlative adjectives. The tag used for superlative adjective is *JJS* by the function *pos_tag*. When a `part of speech` tag is found to be *JJS*, the counter for the superlative adjective is incremented. We believe the use of superlative adjectives is a good practice over the use of intensifiers like very really, fairly etc.
5) **Predeterminers**: Predeterminers are used before determiners or article that gives more information about the noun in the sentence. For example, in the sentences *all these students* and *once a week* the words *all* and *once* are predeterminers. Taguchi et al. have highlighted the importance of linguistic features as an indicator of writing quality and predeterminers play an important role in it [48]. The acronym for a predeterminer is *PDT*. When a `part of speech` tag is found to be *PDT*, the counter for the superlative adjective is incremented.
6) **Coordinating conjunctions**: Coordinating conjunctions are used to join two main clauses. For example, 'My dog Tom has beautiful eyes *but* a notorious personality'. Here *but* is the coordinating conjunction joining two main clauses. The higher frequency of coordinating conjunction used to link sentences plays a significant role in the overall length of paper [49]. As coordinating conjunctions make the sentences larger, we believe larger sentences become harder to understand and leads to low scores. We count the total number of coordinating conjunctions used in the essay. The acronym used for coordinating conjunction is *CC*. When a `part of speech` tag is found to be *CC* for a word by function *pos_tag*, the counter for coordinating conjunction is incremented.
7) **Words ending with -ing**: Nouns and verbs ending with -ing are known as gerunds and participles respectively. Excess use of them makes the writing look naive. Even though using these words is not grammatically wrong, it is advised to choose your word suffix wisely and re-using -ing throughout is condemned [50]. We count the total number of words ending with ing using regular expressions. We use the library called *re*[2] in python. We use regular expression '\b(\w+ing)\b' to identify words ending with -ing and count the occurrences in each sentence and add them to the total counter.
8) **Common sentence length**: Each sentence needs to be of an average length. A sentence too long or too short reflects poor writing and makes the comprehension difficult for the readers [51]. There is an inverse relationship between the grade awarded and difficulty to understand [52]. We believe very long or very short sentences are harder to understand and lead to low scores.
9) **Subject-verb agreement**: Using a singular subject with a plural verb or vice-versa leads to subject-verb disagreement. The total number of singular subjects, plural subjects, and number of different verbs forms are counted. Making these errors are mostly common with non-native english speakers [53].
10) **Repetitive words**: While there are some words that be can safely repeated in a sentence, repeating the same word again and again can distract the grader from the point that writing is trying to make [44]. Being cautious with repetition makes writing more professional. We count the maximum number of repetitions that occurred for a word in an essay. We use the function called *allWorddist* and count the occurrences of most common words using function called *most-common*.
11) **Words**: Too short or a too long essay can reflect on the scores. Counting the total number of words in an essay plays a crucial role on scoring. We count the total number of words used in the essay. We put all the words from the essay into a list and count the length of the essay's word list using the python function *len*. As per the common norm, essay writing tests have a word limit associated where students are asked to not write more or less than certain number of words. We believe using less or more than the word limit affects the scores.
12) **Characters**: Apart from counting the total number of words, it is also important to keep a check on character count as it highlights the total use of alphabets as well as the non-alphabet elements. Total *character count* includes the count of alphabets, punctuation, numbers,

[2] https://docs.python.org/3/library/re.html

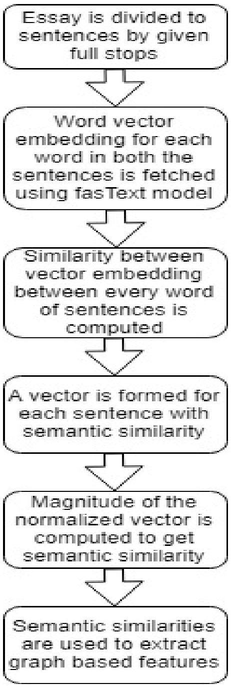

Essay is divided to sentences by given full stops
Word vector embedding for each word in both the sentences is fetched using fasText model
Similarity between vector embedding between every word of sentences is computed
A vector is formed for each sentence with semantic similarity
Magnitude of the normalized vector is computed to get semantic similarity
Semantic similarities are used to extract graph based features

FIGURE 4. A top-down chart describing steps of computing sentence similarities and semantic properties of an essay.

and spaces. We believe if a student uses appropriate number of words but extra spaces or unnecessary punctuation it increases the character count which is not a good practice. In many essay writing contests, a limit is set on the characters to be used and this feature impacts the score. We count the total number of characters used in the essay using the python function *len*.

B. SEMANTIC ATTRIBUTES

Semantics is also known as the study of meaning. The main purpose of any language is to communicate meaning to one another [17]. Semantic attributes are an important aspect of NLP that indicates its meaning. Semantic analysis is a method for extracting and representing the contextual meaning of words or a set of words. Even a very well structured and grammatically correct essay also needs to be well coherent to qualify for good grades. To check the semantics associated with it, we need to make sure the content of the essay does justice to what the question prompt says, and the content flow in the essay's sentences are meaningfully related.

It is crucial that every piece of an essay fits together semantically. Incoherence of a part of the essay with other sub-parts indicates that the particular part is unconnected to the rest of the essay [54]. It is an important criteria during essay evaluation that essay is organized around a central unifying theme [51]. To check the coherence, we compute semantic similarities between the sentences of an essay. We believe comparisons only between consecutive pairs is not enough and we decided to make a semantic comparison between every sentence in the essay. In our research, we propose to compute semantic similarity between not just consecutive statements but also between each pair of sentences to analyze the over all coherence of the essay, Figure 5. The similarities between all the sentences help us derive novel Graph-based features which highlight how different essay sentences are connected semantically and their patterns .

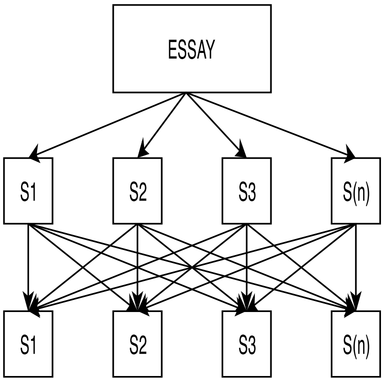


FIGURE 5. Semantic comparisons between each sentence of the essay.

I Computing Semantic Similarities

Word embeddings are mapping of phrases or words in a sentence to vectors [55]. To get the word-vector embeddings, we use embeddings by fastText created by Facebook's AI research lab [56] with *Magnitude* which is a vector utility library [57] and a faster alternative to Gensim [58]. The files with *.magnitude* extension for vector embeddings are designed to allow lazy loading that supports faster look-ups. Lazy loading refers to deferring of initialization of an object until the point at which it is needed. We compute the sentence similarities by providing these vector embeddings to the semantic similarity algorithm [22], [59], [60], Figure 4 and Figure 6. The proposed solution by Pawar *et al.* [22] uses *Wordnet*. *Wordnet* is a lexical database which has words linked together by there semantic relationships. It considers only noun-noun and verb-verb connections. To overcome this and make the algorithm suitable for each `part of speech`, our algorithm considers the similarity between every word of the first given sentence with every word of the second sentence using the vector embedding values. The semantic vectors *V1* and *V2* contain semantic information concerning the words from both the sentences. These two vectors are normalized using their magnitude. The final similarity value is obtained after considering recurrence of words, negation, and Spacy's dependency parser model.

When an essay is given as an input to the system, it is split into sentences by given full stops. We perform pre-processing and convert it to lower-case and remove the stop words. Each sentence from the essay is compared to every other sentence, Figure 5. Each pair of sentences in the essay are tokenized, and similarity between their word vector embedding are computed using *similarity* function by the *Magnitude* library in

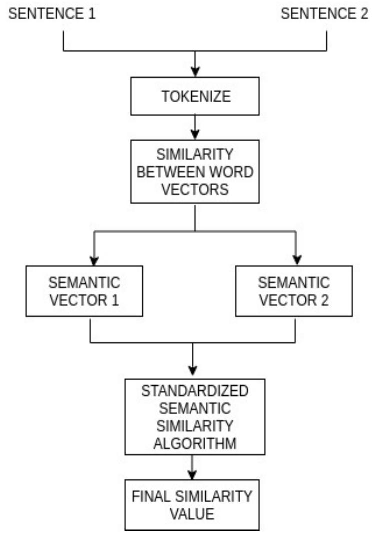


FIGURE 6. Using word vector embedding by fastText with standardized semantic similarity algorithm.

python. A vector is formed for each sentence using *word2vec* function, and these semantic vectors are used by the semantic similarity algorithm, Figure 6. The result of this algorithm is between 0 and 1.

1) EXAMPLE ESSAY

Dear Newspaper, I think that the effects are okay as long as we get off the computers and go outside and see some friends and get some exercise. Computers let us not just talk to each other but it also lets us challenge each other on games without hurting each other it could even stop all ways all at once because we could challenge other countries in war games without killing real living people.We can look up how to stop snake venom from getting to your heart and how to make a how and some arrows to hurt with.It also makes it easier to find health insurance, car insurance, and house insurance. We can check our taxes and stocks.We can look up historical facts on the computer. You can find plumbers, technicians, oil companies, and lumber companies. you can find dates on the computer, too and find information about certain eople too.

This essay is comprised of 8 sentences. Following are the sentences split by given full-stops:

a. Dear Newspaper, I think that the effects are okay as long as we get off the computers and go outside and see some friends and get some exercise.
b. Computers let us not just talk to each other but it also lets us challenge each other on games without hurting each other it could even stop all ways all at once because we could challenge other countries in war games without killing real living people.
c. We can look up how to stop snake venom from getting to your heart and how to make a how and some arrows to hurt with.

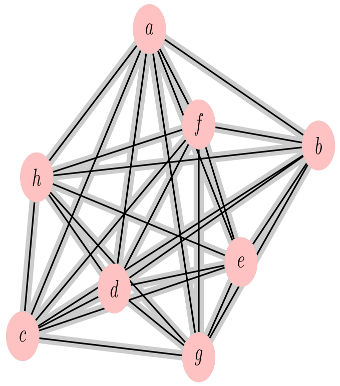


FIGURE 7. Graph based representation of sentences using semantic similarities between sentences as edge weight.

TABLE 1. Semantic similarity values between each sentence.

S	a	b	c	d	e	f	g	h
a	1.0	0.56	0.19	0.59	0.56	0.76	0.56	0.60
b	0.56	1.0	0.34	0.35	0.46	0.55	0.50	0.25
c	0.19	0.34	1.0	0.71	0.52	0.28	0.64	0.32
d	0.59	0.35	0.71	1.0	0.49	0.60	0.69	0.79
e	0.56	0.46	0.52	0.49	1.0	0.57	0.72	0.66
f	0.76	0.55	0.28	0.60	0.57	1.0	0.57	0.61
g	0.56	0.50	0.64	0.69	0.72	0.57	1.0	0.68
h	0.60	0.25	0.32	0.79	0.66	0.61	0.68	1.0

d. It also makes it easier to find health insurance, car insurance, and house insurance.
e. We can check our taxes and stocks.
f. We can look up historical facts on the computer.
g. You can find plumbers, technicians, oil companies, and lumber companies.
h. you can find dates on the computer, too and find information about certain eople too.

Semantic similarity computed between each of these sentences can be seen in the Figure 9 and Table 1. Each of the semantic similarity value ranges from 0 to 1. Each sentence is compared to every other sentence in the essay to check the semantic similarity between them (a value between 0 and 1). Several researchers in the NLP field have used sentences or words transformed to graphs for text summarizing [35], [61]. Deriving motivation from such works, we propose a novel approach to represent semantic similarities between essay sentences as graphs and deriving features from these graphs. Considering each sentence as a vertex and the similarity values as edge weights, the results obtained are transformed into a fully connected graph to view the relations on a spatial space, Figure 7 and Table 1. For some Graph-based features, it is important to use only the strong semantic relations as weak connections can affect the features to a large extent and thus overshadow other powerful connections. To make sure that only the relevant and meaningful connections are considered,

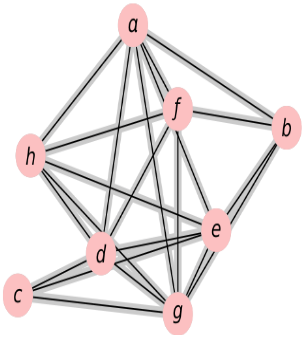


FIGURE 8. Graph based representation of sentences using semantic similarities >0.4 between sentences as edge weight.

TABLE 2. Semantic similarity values >0.4.

S	a	b	c	d	e	f	g	h
a	1.0	0.56	-	0.59	0.56	0.76	0.56	0.60
b	0.56	1.0	-	-	0.46	0.55	0.50	-
c	-	-	1.0	0.71	0.52	-	0.64	-
d	0.59	-	0.71	1.0	0.49	0.60	0.69	0.79
e	0.56	0.46	0.52	0.49	1.0	0.57	0.72	0.66
f	0.76	0.55	-	0.60	0.57	1.0	0.57	0.61
g	0.56	0.50	0.64	0.69	0.72	0.57	1.0	0.68
h	0.60	-	-	0.79	0.66	0.61	0.68	1.0

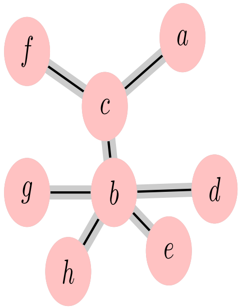


FIGURE 9. Minimum spanning tree.

TABLE 3. Edges used in the tree.

S	a	b	c	d	e	f	g	h
a	-	-	0.19	-	-	-	-	-
b	-	-	0.34	0.35	0.46	-	0.50	0.25
c	0.19	0.34	-	-	-	0.28	-	-
d	-	0.35	-	-	-	-	-	-
e	-	0.46	-	-	-	-	-	-
f	-	-	0.28	-	-	-	-	-
g	-	0.50	-	-	-	-	-	-
h	-	0.25	-	-	-	-	-	-

every connection with a similarity value less than a certain threshold is dropped to obtain Figure 10. See Table 2 to observe every similarity value less than 0.4 has been discarded. After conducting experiments, section IV-D, we observe the system gives best predictions with Graph-based features derived by similarities greater than the threshold of 0.4.

II Graph Based Features

We use semantic similarities represented as graph. The Graph-based features give us more information about how an essay occupies the spatial space. To analyze the structural properties and patterns in a network, we propose graph characteristics as highlighted by Kolaczyk [62]. To obtain the semantic properties of an essay, the following are the Graph-based features computed:

a: MINIMUM SPANNING TREE

Minimum spanning tree is a subset of graph edges that connect all the vertices with minimum possible total weight [63]. When an essay is represented as a graph in a semantic space, it depicts the semantic association between each sentence. The main motivation behind obtaining the minimum spanning tree is to derive the weakest similarity connections in the essay, Figure 9. This graph represents a minimum spanning tree for the fully connected graph in Figure 7. The weakest connections which span through the graph representation of an essay tells us about the minimum coherence, i.e., traversing through all the statements of the essay. We sum the values of the edges weights of the tree and get a minimum spanning tree sum. We convert the semantic similarity results into a sparse matrix using python functionality from *scipy* [64]. This matrix is used to find the minimum spanning tree using the *minimum_spanning_tree* function from *scipy* again.

b: MAXIMUM SPANNING TREE

Maximum spanning tree is a subset of graph edges that connect all the vertices with the maximum possible total weight of the edges. A minimum spanning tree with reciprocated edge weight (for instance, value of 0.56 in cell (a,b) in Table 2 is converted to 1.78 = (1/0.56) in Table 4) is a maximum spanning tree for the original edge weights. Thus, the inverted the values of Figure 7 are used to obtain this sub-graph, Figure 11. Each edge value is reciprocated, see Table 4. As the graph depicts the semantic association between each sentence, to find the most similar connection which binds all the sentences of a graph, we compute the maximum spanning tree. Summing up the non-reciprocated original values from this subset gives us the maximum spanning tree sum.

c: CLOSENESS CENTRALITY

Closeness centrality is measure of centrality in a network [65]. Closeness centrality for a node *a* in the graph is the inverse of the shortest path distances from node *a* to all *n-1* other nodes, while *n* being total number of nodes in the graph. See Equation 1, *d(b,a)*

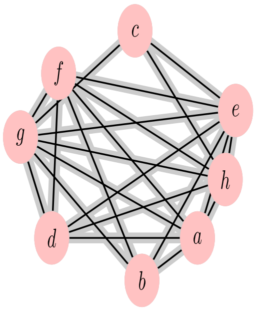


FIGURE 10. Graph based representation of sentences using reciprocated semantic similarities between sentences as edge weight.

TABLE 4. Reciprocated semantic similarities between sentences as edge weight

S	a	b	c	d	e	f	g	h
a	1.0	1.78	5.26	1.69	1.78	1.31	1.78	1.66
b	1.78	1.0	2.94	2.85	2.17	1.81	2	4
c	5.26	2.94	1.0	1.40	1.92	3.57	1.56	3.12
d	1.69	2.85	1.40	1.0	2.04	1.66	1.44	1.26
e	1.78	2.17	1.92	2.04	1.0	1.75	1.38	1.51
f	1.31	1.81	3.57	1.66	1.75	1.0	1.754	1.63
g	1.78	2	1.56	1.44	1.38	1.75	1.0	1.47
h	1.66	4	3.12	1.26	1.51	1.63	1.47	1.0

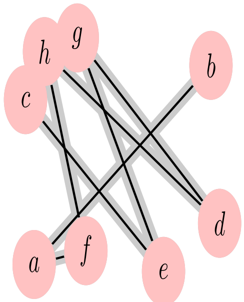


FIGURE 11. Maximum spanning tree.

TABLE 5. Original edge weight used in the tree.

S	a	b	c	d	e	f	g	h
a	-	0.56	-	-	-	0.76	-	-
b	0.56	-	-	-	-	-	-	-
c	-	-	-	-	0.52	-	-	-
d	-	-	-	-	-	-	0.69	0.79
e	-	-	0.52	-	-	-	0.72	-
f	0.76	-	-	-	-	-	-	0.61
g	-	-	-	0.69	0.72	-	-	-
h	-	-	-	0.79	-	0.61	-	-

represents shortest path distance between a and b and n is the number of nodes that can reach the node a. The distance d here represents the semantic similarity values as edge weights.

$$Closeness\_centrality(a) = \frac{1}{\sum_{b=1}^{n-1} d(b,a)} \quad (1)$$

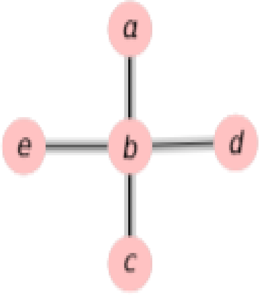


FIGURE 12. Graph with 5 nodes.

TABLE 6. Centrality values for each node.

Node	Centrality
a	0.14
b	0.25
c	0.14
d	0.14
e	0.14
Average	0.162

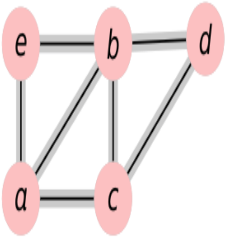


FIGURE 13. Graph with 5 nodes.

For each graph based essay representation we find closeness centrality for each node. The Figure 12 shows a graph with 5 nodes with 4 edges and Table 6 displaying closeness centrality values when assumed that distance between each node is 1. The average centrality of all the nodes for this graph is 0.16. The Figure 13 and Table 7 show another graph with 5 nodes with 7 edges and the average closeness centrality of 0.41. We want to emphasize that the second graph has higher average centrality value, thus each node in the graph is more central and has more cohesion associated to the essay.

d: GRAPH ECCENTRICITIES

The maximum distance between a vertex to all other vertices in the graph is called eccentricity of the vertex. The eccentricity of a node a is the maximum distance from a to all other nodes in the graph G. The graph in Figure 14, has seven nodes and the Table 8 displays eccentricity for each node. Eccentricities can be further studied as:

- Diameter: The diameter of a graph is the greatest distance between any pair of vertices. From

TABLE 7. Centrality values for each node.

Node	Centrality
a	0.14
b	1.0
c	0.14
d	0.14
e	0.66
Average	0.416

TABLE 8. Node eccentricities.

Node	Eccentricity
a	3
b	2
c	3
d	4
e	3
f	3
g	4

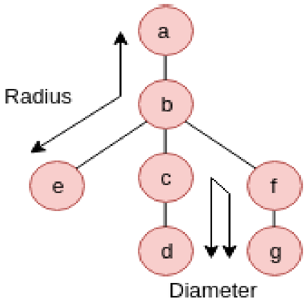


FIGURE 14. Graph with 5 nodes.

all the eccentricities of the vertices in a graph, the diameter of the connected graph is the maximum of all those eccentricities. The diameter of a graph indicates how widespread the nodes are. A large diameter value indicates a widespread graph which is less cohesive [66]. In Figure 14, the maximum eccentricity values is 4 for node *d*. Thus, the diameter of the graph is 4.

- Radius: The minimum eccentricity from all the vertices is considered as the radius of the graph. From all the eccentricities of the vertices in a graph, the radius of the connected graph is the minimum of all those eccentricities. The minimum eccentricity for a node in Figure 14 is 2. Therefore, the radius of the graph is 2.

e: DENSITY DIFFERENCE

Density of graph can be defined as the actual number of edges compared to the possible number of edges of that graph. A higher number of edges in the graph indicates a higher density. The Equation 2 is used to derive the density of a graph, m is the number of edges and n is the number of nodes. We compare the graph in Figure 7 with the graph in Figure 8 to compute the density difference before and after dropping all the similarities below 0.4. This difference highlights the number of weak connections with similarities less than 0.4 in the graph, higher density difference indicates a high number of weak connections existed in the graph before we dropped them.

$$d = \frac{2m}{n(n-1)} \tag{2}$$

f: NUMBER OF CENTRAL NODES

Nodes with eccentricity equal to the radius (which is the minimum eccentricity possible between any pair of vertices) are referred to as central nodes. More central number of nodes implies a closely related compact graph. In Figure 14, only one node, i.e., b has eccentricity equal to the radius of the graph. Thus, the number of central nodes for this graph is 1.

g: SIGNIFICANT WORDS AND THEIR SIMILARITY

As per existing researches in the literature, closeness between the essay and the question prompt is important [67].

We use TF-IDF to find important words in the essay and the prompt. *TF-IDF* stands for Term frequency - Inverse Data frequency. For each sentence in the essay, we compute the frequency of each term in the sentence that is called *TF*. *IDF* refers to the importance of the word's existence in a text, the words which occur rarely are weighed up and which occur too often are weighed down [68].

$$tf(t,d) = f_{t,d} \tag{3}$$

In Equation 3, the term frequency *tf(t,d)* is the number of times that a term t appears in one sentence d.

$$idf(t,D) = \log \frac{|D|}{n_t} \tag{4}$$

In Equation 4, the inverse document frequency *idf(t, D)* looks at whether the term t is common or not across all the set of sentences in the essay D. n_t is the total number of sentences containing the term t.

$$\mathrm{tf} - \mathrm{idf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D) \tag{5}$$

In Equation 5, *tf-idf* shows the importance of the term t in a sentence d given the essay D. We filter nouns and verbs out of the essay and the question prompt and compute TF-IDF. We find important words for each sentence in the essay and consolidate them and then pick up top 10 from each essay based on their *tf-idf* scores. Top 10 words are picked based on the *TF-IDF* from both the essay and the question prompt. A limitation can occur if an essay prompt consists of less than 10 words which can be handled by making

TABLE 9. **Polarity values of same sentence under different contexts.**

Sentence	Values of Sentiment based features
This car is good	Negative : 0, Positive: 0.492, Neutral : 0.508
This car is good!	Negative : 0, Positive: 0.433, Neutral : 0.567
This car is good!!!	Negative : 0, Positive: 0.486, Neutral : 0.514
This car is GOOD!	Negative : 0, Positive: 0.472, Neutral : 0.528
This car is extremely good	Negative : 0, Positive: 0.492, Neutral : 0.508
This car is marginally good	Negative : 0, Positive: 0.442, Neutral : 0.458
This car is good, but it's fuel economy is bad	Negative : 0.31, Positive: 0.192, Neutral : 0.558
This car is not good	Negative : 0.376, Positive: 0, Neutral : 0.624

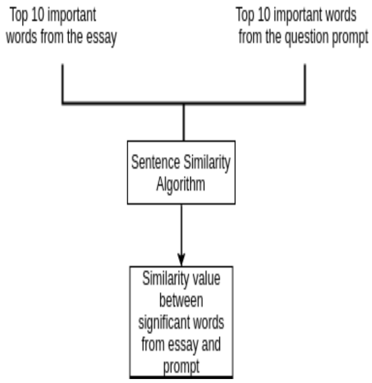

Top 10 important words from the essay
Top 10 important words from the question prompt
Sentence Similarity Algorithm
Similarity value between significant words from essay and prompt

FIGURE 15. **Finding significant words and computing their similarity.**

changes in the code. These two-word sets are compared to each other using the semantic similarity algorithm with fastText word embeddings which gives a similarity value for both the lists. Figure 15 is a chart showing the process of similarity derivation between top-words from essay and prompt.

h: SIMILARITY BETWEEN THE PROMPT AND THE ENTIRE ESSAY

Semantic similarity between the entire essay and the question prompt is computed to find how meaningfully similar they are. Since every essay is written as a response to a question prompt, we believe they should display semantic similarity between them. We use the semantic similarity algorithm as described in Section III-B to compute semantic similarity between the essay response and question prompt.

C. SENTIMENT ATTRIBUTES

Sentiment Analysis is the study of opinions, attitudes, and emotions toward an entity [69]. Each writer has a unique or changing tone towards the subject being written about [70]. For example, if a writer wants to write about his disagreement about a scenario, this analysis tells us about how negative the language used is or how positive or neutral the tone of the writer is. Sentimental analysis plays an important role in AES. The factors making it important are listed below:

1) In argumentative and persuasive essays as an author needs to defend and prove his/her point of view on the subject, their tone and the way of textual sentiment expression affect their writing [71].
2) The sentiment analysis of figurative speech in the essays helps us know the polarity inclination they contribute to the text which is otherwise difficult to comprehend by the computer [72].

A study on existing sentiment analysis methods [73] show that VADER is one of the best performing open-source sentiment analysis tool. Valence Aware Dictionary and sentiment Reasoner (VADER), which is a rule-based model for sentiment analysis [74]. It is fully open-sourced under the MIT License. It does not require any training data but is constructed from a valence-based, human-curated gold standard sentiment lexicon. It uses a lexicon of words already trained as per their sentimental inclination as positive, negative or neutral. VADER uses Amazon's Mechanical Turk to get scores for the lexicon [75]. The polarity of each sentence is checked, and final polarity values towards being positive, negative and neutral are obtained in terms of percentage. The positive, negative and neutral scores represent the chunk of text that falls in these categories. This means if our essay was rated as 50 percent Positive, 25 percent Neutral and 25 percent Negative, these values should add to a 100 percent.

Following are the important key points used by VADER to analyze sentiments, refer Table 9 for the polarity values given by VADER to the sentences:

- Punctuation
- Using upper case
- Degree modifiers
- Conjunctions
- Preceding Tri-gram

IV. IMPLEMENTATION AND EVALUATION

A. DATA

To validate and compare our results to the exsiting systems, we use data from the Automated Student Assessment

TABLE 10. ASAP data-set.

Set	Essays	Genre	Avg. Length	Score
1	1,783	Arg	350	2-12
2	1800	Arg	350	1-6
3	1,726	Res	150	0-3
4	1,772	Res	150	0-3
5	1,805	Res	150	0-4
6	1,800	Res	150	0-4
7	1,569	Nar/Per/Exp	250	0-30
8	723	Nar	650	0-60

Prize (ASAP)[3] competition sponsored by the William and Flora Hewlett Foundation (Hewlett) held in 2012. The dataset is composed of 8 different data-sets of different genres which are argumentative (Arg), responsive (Res), narrative (Nar), persuasive (Per) and expository (Exp). Each data-set is a collection of responses to its own prompt. Students from grade 7 to grade 10 have written the essays ranging from 150 to 550 words per essay, refer Table 10. A minimum of 2 human graders has provided the grades which are available to us.

B. EVALUATION METRIC

The evaluation metric used to test our system is Quadratic Weighted Kappa (QWK), as this was the official evaluation metric chosen by the ASAP competition. QWK is a measure of agreement between two raters. In case of essay evaluation system it is the agreement between predicted score by the system and the grade by human rater. QWK ranges between values 0 (no agreement) to 1 (complete agreement) [76], refer Table 11. Each dataset *E* has *N* number of possible ratings. Every essay in each data-set can be indicated by a tuple (*ea,eb*) where *ea* refers to the human rater's score and *eb* refers to the predicted score by the evaluation system. An *N-by-N* histogram matrix *O* is constructed over the essay ratings, where *Oi,j* refers to the number of essays with grade *i* by human grader and a grade *j* by prediction system. An *N-by-N* matrix of weights *w*, is calculated based on the difference between rater's scores refer Equation 6 and Quadratic Weighted Kappa *k* is found by the Equation 7.

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{6}$$

$$k = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \tag{7}$$

C. FEATURE SELECTION

Feature selection is the core building block of a machine learning model and has a huge impact on the performance. Moderately performing or irrelevant features can negatively affect the prediction. To make the best use of a machine learning model, the foremost significance should be given to feature selection and data pre-processing. Even a single bad feature can hamper the model's performance. Some key benefits of feature selection are:

TABLE 11. QWK value interpretations.

QWK	Agreement strength
<0.2	Poor
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Good
0.81-1.00	Very Good

- *Reduces over-fitting*: Getting rid of some features helps to get rid of redundant data. Thus the machine learning model's prediction is not based on any noisy data.
- *Reduces training time*: Fewer features means less complexity and lesser training time for the algorithm.
- *Enhances the performance of the prediction model*: When only the best predicting features are fed to the model, the modeling accuracy improves.

To select the best features we perform selection techniques [77] mentioned below:

1) UNIVARIATE SELECTION

We use a univariate linear regression test by sci-kit-learn [78]. A linear model is used to derive the effect of each feature (regressor) on the predictability of the model. Univariate feature selection works by selecting the best predictive features based on statistical tests. As we are solving regression problem here we use the function *f_regression* with *sklearn.feature_selection* from skicit learn library in python [78].

2) RECURSIVE FEATURE ELIMINATION

Important features are selected by recursively removing features and testing the prediction with remaining ones to find which features have with most predictive power. This is done using *featureimportances* function by *skicit-learn* [78] in python.

In Table 12, the second and third column displays the results from Univariate linear regression test and ranks from Recursive feature elimination respectively.

3) QWK VS. FEATURES

As we are trying to minimize the number of features for better prediction, it is important to decide the minimum number of features possible without compromising the performance of the model. As per the ranking obtained by recursive feature elimination, we keep adding features to the model starting from the top ranking feature, i.e., the number of characters and recursively adding second, third and so on till the last one, i.e., the diameter of the graph. We derive the QWK value after each iteration of adding a new feature to the model

[3] https://www.kaggle.com/c/asap-aes

TABLE 12. Results from feature selection techniques.

Feature name	Univariate selection	Ranking from recursive feature elimination
Number of characters	3.12E+02	1
Number of words	2.91E+02	2
Semantic similarity between top-words of essay and prompt	2.20E+02	3
Sum of maximum spanning tree	1.67E+02	4
Sum of minimum spanning tree	1.69E+02	5
Density difference between original and reduced graph of similarities	1.63E+02	6
Common sentence length	1.63E+00	7
Unique part of speeches used	1.58E+02	8
Words ending with -ing	1.25E+02	9
Misspelled words	1.25E+02	10
Number of singular subjects	2.07E+02	11
Closeness centrality	1.20E+02	12
Number of plural subjects	8.84E+01	13
Number of co-ordinating conjunctions	8.29E+01	14
Past tense verb	7.21E+01	15
Verb, base-form	6.62E+01	16
Verb, present participle	8.61E+01	17
Verb, non-3rd person singular present	8.84E+01	18
Positive polarity	7.59E+01	19
Verb, past participle	8.29E+01	20
Negative polarity	4.12E+01	21
Number of most frequent repetitive word	2.22E+01	22
Neutral polarity	6.19E+00	23
Semantic similarity to question prompt	2.09E+01	24
Number of superlative adjectives	2.09E+00	25
Number of predeterminants	1.65E-01	26
Number of exesntial-there	3.52E-01	27
Radius	2.68E-01	28
Central nodes	8.80E+00	29
Diameter	8.80E+00	30

for each of the eight data-sets. A graph is plotted using the average QWK values over all data-sets vs. the number of features used towards obtaining it, Figure 16. We see the top 23 ranked features from the recursive feature elimination technique gives us the best performance with highest QWK value. Therefore, we use the top 23 features from the Table 12 for our final prediction model.

4) PREDICTABILITY OF EACH FEATURE

ASAP data-set has eight data-sets of different genres. Therefore it is important to test how the individual features behave in terms of prediction over different data-sets. We run our prediction model for each feature individually over each data-set recursively and obtain a QWK value for each. This helps us to validate the above-mentioned feature elimination techniques and help us select the features which perform well over each data-set. Fig. 17 shows the top 23 features selected for the model have a good performance in term of prediction power on each of the data-sets.

Based on the results mentioned above, we selected the top 23 best performing features for our prediction model. We obtain results from univariate feature selection and ranking from the recursive feature elimination technique. To validate these techniques, we use the QWK values for additional assessment. We plot a graph of QWK values vs. the number of features which are recursively added to the model based on ranks from recursive feature elimination. This plot shows the model performs the best when the top 23 features are used. Additionally, to analyze each feature's prediction power we

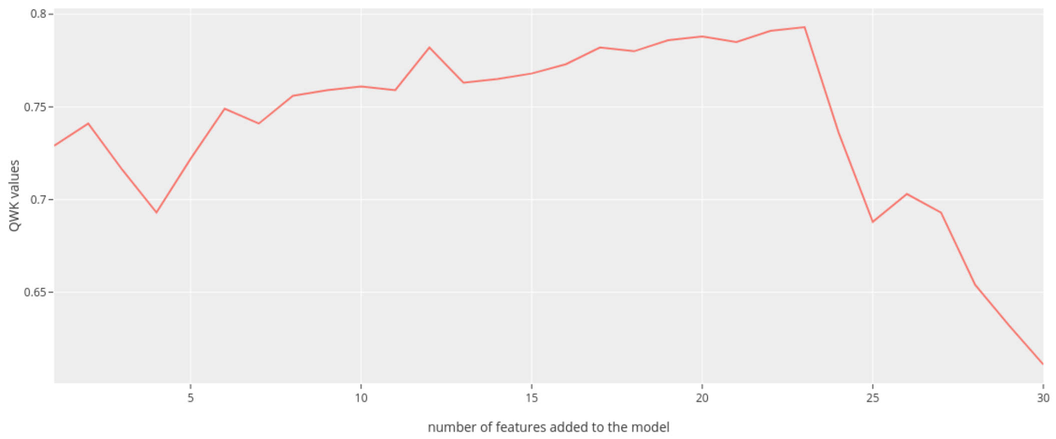
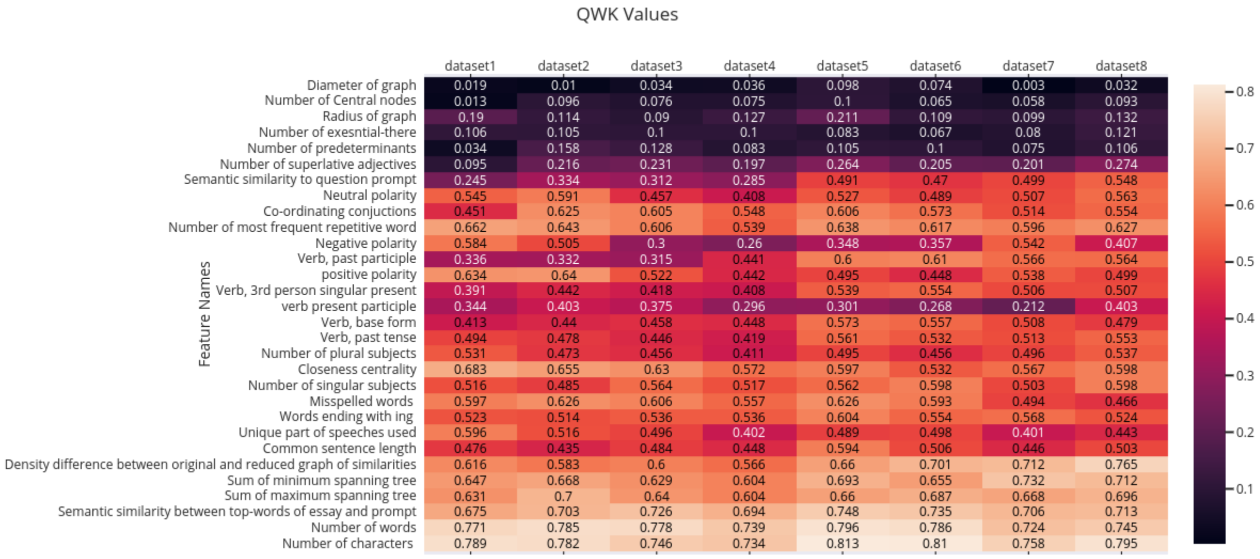


FIGURE 16. QWK values vs number of features recursively added to the model.



	dataset1	dataset2	dataset3	dataset4	dataset5	dataset6	dataset7	dataset8
Diameter of graph	0.019	0.01	0.034	0.036	0.098	0.074	0.003	0.032
Number of Central nodes	0.013	0.096	0.076	0.075	0.1	0.065	0.058	0.093
Radius of graph	0.19	0.114	0.09	0.127	0.211	0.109	0.099	0.132
Number of exesntial-there	0.106	0.105	0.1	0.1	0.083	0.067	0.08	0.121
Number of predeterminants	0.034	0.158	0.128	0.083	0.105	0.1	0.075	0.106
Number of superlative adjectives	0.095	0.216	0.231	0.197	0.264	0.205	0.201	0.274
Semantic similarity to question prompt	0.245	0.334	0.312	0.285	0.491	0.47	0.499	0.548
Neutral polarity	0.545	0.591	0.457	0.408	0.527	0.489	0.507	0.563
Co-ordinating conjuctions	0.451	0.625	0.605	0.548	0.606	0.573	0.514	0.554
Number of most frequent repetitive word	0.662	0.643	0.606	0.539	0.638	0.617	0.596	0.627
Negative polarity	0.584	0.505	0.3	0.26	0.348	0.357	0.542	0.407
Verb, past participle	0.336	0.332	0.315	0.441	0.6	0.61	0.566	0.564
positive polarity	0.634	0.64	0.522	0.442	0.495	0.448	0.538	0.499
Verb, 3rd person singular present	0.391	0.442	0.418	0.408	0.539	0.554	0.506	0.507
verb present participle	0.344	0.403	0.375	0.296	0.301	0.268	0.212	0.403
Verb, base form	0.413	0.44	0.458	0.448	0.573	0.557	0.508	0.479
Verb, past tense	0.494	0.478	0.446	0.419	0.561	0.532	0.513	0.553
Number of plural subjects	0.531	0.473	0.456	0.411	0.495	0.456	0.496	0.537
Closeness centrality	0.683	0.655	0.63	0.572	0.597	0.532	0.567	0.598
Number of singular subjects	0.516	0.485	0.564	0.517	0.562	0.598	0.503	0.598
Misspelled words	0.597	0.626	0.606	0.557	0.626	0.593	0.494	0.466
Words ending with ing	0.523	0.514	0.536	0.536	0.604	0.554	0.568	0.524
Unique part of speeches used	0.596	0.516	0.496	0.402	0.489	0.498	0.401	0.443
Common sentence length	0.476	0.435	0.484	0.448	0.594	0.506	0.446	0.503
Density difference between original and reduced graph of similarities	0.616	0.583	0.6	0.566	0.66	0.701	0.712	0.765
Sum of minimum spanning tree	0.647	0.668	0.629	0.604	0.693	0.655	0.732	0.712
Sum of maximum spanning tree	0.631	0.7	0.64	0.604	0.66	0.687	0.668	0.696
Semantic similarity between top-words of essay and prompt	0.675	0.703	0.726	0.694	0.748	0.735	0.706	0.713
Number of words	0.771	0.785	0.778	0.739	0.796	0.786	0.724	0.745
Number of characters	0.789	0.782	0.746	0.734	0.813	0.81	0.758	0.795

FIGURE 17. Heatmap of QWK values for each feature over the 8 data-sets.

recursively run the model with just one feature at a time over all data-sets and obtain a heatmap to identify unsatisfactorily performing features. This heatmap validates the performance of our selected top 23 features. Thus, they are added to the prediction model.

D. THRESHOLDING FOR GRAPH-BASED FEATURES

As we discussed about the Graph-based feature extraction from semantic similarities in the essay after dropping the similarities less than a certain threshold. In this section, we elaborate on the selection of this threshold value.

For Graph-based features like closeness centrality, density difference between original and reduced graph, sum of minimum and maximum spanning trees, it is important to use only the strong semantic relations as even one weak connection can affect the predictions. We checked how the system's predictability behaves with Graph-based features including all edges compared to the predictability after dropping edges with different thresholds values.

We obtain different sets of graph based features with semantic similarities greater than threshold values starting from 0.1 till 0.9. We test our prediction model with these features and find QWK value for every data-set for each set of features. We found that feature set obtained with threshold of 0.4 gives us the best prediction values, Table 13.

E. PREDICTION MODELS

We treat score prediction as a regression problem and use supervised learning for it. We chose a various learning algorithms and train the scored essays by humans and then provide it with features associated with unseen essays to obtain the predicted score using the regression function. The technologies and libraries used for prediction models are sci-kit-

TABLE 13. Comparison of results considering different threshold to derive semantic similarities.

Threshold	D-1	D-2	D-3	D-4	D-5	D-6	D-7	D-8	Avg.
0.1	0.79	0.79	0.56	0.59	0.85	0.69	0.68	0.70	0.71
0.2	0.78	0.63	0.78	0.55	0.77	0.66	0.70	0.65	0.69
0.3	0.80	0.77	0.62	0.70	0.78	0.80	0.71	0.61	0.72
0.4	**0.83**	**0.81**	**0.77**	**0.72**	**0.87**	**0.82**	**0.75**	**0.78**	**0.79**
0.5	0.79	0.69	0.66	0.66	0.77	0.78	0.74	0.73	0.73
0.6	0.88	0.60	0.61	0.63	0.78	0.66	0.74	0.63	0.69
0.7	0.87	0.60	0.60	0.58	0.76	0.84	0.74	0.61	0.70
0.8	0.78	0.60	0.61	0.55	0.75	0.68	0.73	0.63	0.67

TABLE 14. Support vector regressor parameters.

Parameter	Value
kernel	sigmoid

TABLE 15. Random forest regressor parameters.

Parameter	Value
n_estimator	1000
random_state	42

learn [78], Keras [79], and Numpy [80]. As there are many supervised regression models available, it is crucial to find which one suits best for our problem and yields good results. The three prediction models we tried:

1) SUPPORT VECTOR MACHINE

The method provided by support vector machines to handle regression problems is called support vector regression [81]. A kernel function is a method to solve the non-linear problem by a linear classifier. We use the sigmoid kernel, which is equal to a using a two-layer, perceptron neural network and found to perform well in regression problems [82]. We use the SVR function from the scikit-learn library in python to run this prediction model. Refer Table 14 for parameters used.

2) RANDOM FOREST REGRESSOR

Random forest is ensemble of multiple decision trees. Rather than depending on a single decision tree, it depends on various decision trees on sub-samples of the data-set [83]. We use the *RandomForestRegressor* method from the scikit-learn library in python to run this prediction model. Refer Table 15 for the value of parameters used in the model. After running experiments with several combinations of parameter values, it was found the model worked the best with the mentioned values in Table 15.

3) THREE LAYER NEURAL NETWORK

We use Keras API which runs high-level deep neural networks [79]. The neural network consist of three layers with 23 inputs, 2 hidden layers and one output layer:

TABLE 16. Three layer neural network parameters.

Parameter	Value
input_dim	23
kernel_initializer	normal
activation	relu
optimizer	adam
loss	mean_squared_error
epochs	100
batch_size	50

- Input layer: This layer which feeds the input data to the model. The input layer has the same number of neurons as input attributes, i.e., 23 in our case.
- Hidden layer: These layers use *backpropagation* to optimize the weights of the input variables thus improving the prediction power. We use two hidden layers in our model. As per experiments conducted we observed that adding anymore number of layers to the model do not help improve the results any further; therefore, we use two hidden layers.
- Output layer: This layer gives the final output based on inputs and the hidden layers.

We standardize our data as they all vary in their scale. Refer Table 16 for other parameters used in running the model.

We run these three prediction models over all the eight data-sets from ASAP. Table 18 shows the QWK scores obtained over different data-sets with these models. We observe using three layered neural network in scikit-learn with the KerasRegressor class wrapper performs the best in terms of score predictions. Thus, we choose the three layered neural network as our final prediction models for our system's results.

F. EFFECT OF SYNTACTIC, SEMANTIC AND SENTIMENT BASED FEATURES

As the title of the article suggest, we propose to use the syntactic, semantic and sentiment based features together to help automate the process of essay's score predictions. To justify their use together, we conducted experiments using these three set of features separately, using two sets at a time

TABLE 17. Effect of syntactic, semantic and sentiment based features.

Feature Set	**Average QWK over 8 data sets**
Syntax	0.701
Semantic	0.653
Sentiment	0.392
Syntax with Semantic	0.741
Syntax with Sentiment	0.744
Semantic with Sentiment	0.679
Syntactic, Sentiment and Semantic	0.793

TABLE 18. Comparison of QWK for different supervised learning models in score prediction.

Learning Models	**D-1**	**D-2**	**D-3**	**D-4**	**D-5**	**D-6**	**D-7**	**D-8**
SVM (sigmoid kernel)	0.78	0.55	0.61	0.64	0.69	0.61	0.73	0.65
Random forest	0.77	0.54	0.57	0.51	0.75	0.61	0.71	0.66
Three layer neural network	0.83	0.81	0.77	0.72	0.87	0.82	0.75	0.78

TABLE 19. Comparison of results with existing systems.

System	**No. of Features**	**D-1**	**D-2**	**D-3**	**D-4**	**D-5**	**D-6**	**D-7**	**D-8**	**Avg.**
The Proposed System	**23**	0.83	**0.81**	0.77	0.72	**0.87**	**0.82**	0.75	**0.78**	0.793
SBLSTMA	14 plus several sub-features	**0.86**	0.73	**0.78**	**0.82**	0.84	0.82	0.81	0.75	0.80
SVMrank [84]	33	0.80	0.68	0.67	0.73	0.80	0.71	0.77	0.71	0.73
SKIPFLOW [85]	14 plus several sub-features	0.83	0.68	0.69	0.79	0.81	0.810	0.800	0.69	0.76
Tpioc-BiLSTM-attention [86]	Not mentioned	0.82	0.69	0.69	0.81	0.81	0.82	0.80	0.70	0.77
e-rater [9]	46	0.82	0.69	0.72	0.80	0.81	0.75	**0.81**	0.70	0.77
IntelliMetric [8]	400	0.78	0.68	0.73	0.79	0.83	0.76	**0.81**	0.68	0.76
BLRR [67]	15 plus several sub-features	0.76	0.60	0.62	0.74	0.78	0.77	0.73	0.62	0.70
TDNN [87]	Not clear	0.76	0.68	0.62	0.75	0.73	0.67	0.65	0.57	0.68

and then all three together. We observe when we use all three sets of these features together our system performs the best in terms of score predictions. We run the selected prediction model over all 8 data-sets and average the performance of each set of features by obtaining QWK values, Table 17 shows how the average QWK changes over different combinations of features. We get the best results when all three set of features i.e., syntactic, semantic and sentiment are used together.

G. RESULTS AND DISCUSSIONS

To evaluate the automated evaluation system's predictive power, QWK for the predicted scores of each data-set is calculated. Only the top 23 most significant features are used, Table 12. Treating score prediction as a regression problem, we use a Keras based regression model on an i5 processor with 16GB RAM and 1050 Ti graphics card. We divide the data into using a ratio of 75:25 and set aside the 25% for validation of the model later. We use 75% of the data for training the model with ten-fold cross-validation to validate the model's performance. We use the remaining 25% which is never seen by the model during training to test the score predictions by the trained model. We tested the model for each of the eight data-sets to obtain a QWK value. We conducted experiments with different parameter values for neural networks and found the best results when the number of epochs is equal to 100, and the batch size is equal to 50 for the regression model. Table 19 shows the comparison between QWK values over each data-set from our proposed system compared to popular existing automated scoring systems. The results of other systems being compared were obtained from literature [8], [84] and Kaggle's website. The proposed system gives improved results with a fewer number of features involved, while covering all the necessary aspects of language processing, making it very reliable for essay grading with uniform assessment thoroughly. Our system uses only 23 features, which is significantly less than the number used by all the other systems in comparison which reduces the noise during model training, refer Table 19. Our system performs better than any other system in four out of eight data-sets in comparison, with an average QWK of 0.793. The only system outperforming our system in the remaining four data-sets is SBLSTMA [84] with average QWK value of 0.801, i.e., only 0.8% better than our system. SKIPFLOW [85] and SBLSTMA uses 14 main features, plus many more sub-features, which have not been mentioned explicitly in the published research, thus, making the research non-reproducible and making it hard to make a comparison. We try to contact the authors to provide information about sub-features used in their research, but there was no response. The system Tpioc-BiLSTM-attention [86] which works via hierarchical recurrent model does not provide any details about the features used in their published research. We also want to emphasize that extraction of a massive number of features vs. 23 features adds to time complexity as well. We also compared our system to work published recently in 2018; the system is called TDNN [87] which uses a two-layer neural network to reach an average QWK score of 0.7365, i.e., 7.1% lesser than us. To the best of our knowledge, our system uses the minimum number of features compared to existing systems with better results.

V. CONCLUSION & FUTURE WORK

Our proposed system successfully incorporates not only the rule-based grammar and syntax checks, but also the semantic similarities within the essay depicting its coherence. We propose to use semantics based graph based relationships within the essay content and polarity of opinion expressions. We incorporate pragmatic syntax features, semantic features depicting coherence based on accurate semantic similarity values, and sentiment-based polarity features. Thus, our research will help to reduce the number of independent features needed to be extracted from the text while utilizing the most vital features needed in automated essay evaluation for better prediction accuracy. Lesser is the number of features lesser is the redundant data; thus, predictions are not over-fitted. Our work not only provides accuracy values but also provides details to the readers making it reproducible. As compared to other existing systems, our work is more transparent and repeatable. Thus, this research can help eradicate the hours of manual work for teachers, giving them the freedom to concentrate more on academic teaching and learning and also helps give students the assurance of fair and consistent assessment throughout every submission. There are other machine learning models which can be explored and tested with the proposed system in this research. Models like LSTM [88] seem promising for sequential data if work is done to derive the correlation between essay scores in the data-set LSTM can give good results. Existing researchers state that semantics of a coherent essay changes gradually through its text [13]. This approach encourages comparison of consecutive sentences in the essay to study the information flow. Many times, as a matter of writing style, abrupt shifts between consecutive sentences are used to convey information in successive sentences [89]. To study the coherence and flow of information in an essay, a comparison between sentences can be done using a sliding window frame. An optimized value for the sliding window can be obtained, and the sentences which happen to fall within the window can be compared for semantic similarity to assess the value of information. We believe this model lacks the study of ontology-based connections in the essay's text and careful extraction of ontology-based features can be beneficial [90]. The model can be further improved by including other types of centrality based features existing in the graph networks [91].

ACKNOWLEDGEMENT

The cost of open access fee is covered by NSERC Discovery Grant, held by Vijay Mago. The computational resources are provided by the Department of Computer Science at Lakehead University.

REFERENCES

[1] M. D. Shermis and J. C. Burstein, *Automated Essay Scoring: A Cross-Disciplinary Perspective*. Evanston, IL, USA: Routledge, 2003.

[2] E. B. Page, "The imminence of... Grading essays by computer," *Phi Delta Kappan*, vol. 47, no. 5, pp. 238–243, 1966.

[3] E. B. Page, "The use of the computer in analyzing student essays," *Int. Rev. Edu.*, vol. 14, no. 2, pp. 210–225, 1968.

[4] D. Higgins, J. Burstein, D. Marcu, and C. Gentile, "Evaluating multiple aspects of coherence in student essays," in *Proc. HLT-NAACL*, 2004, pp. 185–192.

[5] T. Miller, "Essay assessment with latent semantic analysis," *J. Educ. Comput. Res.*, vol. 29, no. 4, pp. 495–512, 2003.

[6] P. Foltz, L. Streeter, K. Lochbaum, and T. Landauer, "Implementation and applications of the intelligent essay as-sessor," in *Handbook of Automatted Essay Evaluation*, M. Shermis and J. Burstein, Eds. New York, NY, USA: Routledge, 2013, pp. 68–88.

[7] T. K. Landauer, D. S. McNamara, and S. Dennis, *Handbook of Latent Semantic Analysis*. Sussex, U.K.: Psychology Press, 2013.

[8] M. T. Schultz, "The IntelliMetric automated essay scoring engine—A review and an application to Chinese essay scoring," in *Handbook of Automated Essay Scoring: Current Applications and Future Directions*, M. D. Shermis and J. Burstein, Eds. New York, NY, USA: Routledge, 2013, pp. 89–98.

[9] J. Burstein, *The E-Rater Scoring Engine: Automated Essay Scoring With Natural Language Processing*. 2003.

[10] S. M. Lottridge, E. M. Schulz, and H. C. Mitzel, "Using automated scoring to monitor reader performance and detect reader drift in essay scoring," in *Handbook of Automated Essay Evaluation: Current Applications and New Direction*, M. D. Shermis and J. Burstein, Eds. New York, NY, USA: Routledge, 2013, pp. 233–250.

[11] P. W. Foltz, W. Kintsch, and T. K. Landauer, "The measurement of textual coherence with latent semantic analysis," *Discourse Processes*, vol. 25, nos. 2–3, pp. 285–307, 1998. doi: 10.1080/01638539809545029.

[12] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Conf. Uncertainty Artif. Intell.*. San Mateo, CA, USA: Morgan Kaufmann, 1999, pp. 289–296.

[13] P. W. Foltz, W. Kintsch, and T. K. Landauer, "The measurement of textual coherence with latent semantic analysis," *Discourse Processes*, vol. 25, nos. 2–3, pp. 167–184, 1998. doi: 10.1080/01638539809545029.

[14] B. B. Klebanov, N. Madnani, J. Burstein, and S. Somasundaran, "Content importance models for scoring writing from sources," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2014, pp. 247–252.

[15] K. Zupanc and Z. Bosnić, "Automated essay evaluation with semantic analysis," *Knowl.-Based Syst.*, vol. 120, pp. 118–132, Mar. 2017.

[16] M. Chein and M.-L. Mugnier, *Graph-Based Knowledge Representation: Computational Foundations of Conceptual Graphs*. Springer, 2008.

[17] C. Goddard, *Semantic Analysis: A Practical Introduction*. London, U.K.: Oxford Univ. Press, 2011.

[18] C. T. Meadow, B. R. Boyce, and D. H. Kraft, *Text Information Retrieval Systems*, vol. 20. San Diego, CA, USA: Academic, 1992.

[19] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 8, pp. 1138–1150, Aug. 2006.

[20] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using Web search engines," in *Proc. WWW*, vol. 7, 2007, pp. 757–766.

[21] Z. He, S. Gao, L. Xiao, D. Liu, H. He, and D. Barber, "Wider and deeper, cheaper and faster: Tensorized LSTMs for sequence learning," in *Adv. neural Inf. Process. Syst.*, 2017, pp. 1–11.

[22] A. Pawar and V. Mago, "Challenging the boundaries of unsupervised learning for semantic similarity," *IEEE Access*, vol. 7, pp. 16291–16308, 2019.

[23] H. Rubenstein and J. B. Goodenough, "Contextual correlates of synonymy," *Commun. ACM*, vol. 8, no. 10, pp. 627–633, Oct. 1965.

[24] B. Liu, "Sentiment analysis and opinion mining," *Synth. Lectures Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[25] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *J. Comput. Sci.*, vol. 2, no. 1, pp. 1–8, Mar. 2011.

[26] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proc. 42nd Annu. Meeting Assoc. Comput. Linguistics*, 2004, p. 271.

[27] K. Robinson and V. Mago, "Birds of prey: Identifying lexical irregularities in spam on twitter," in *Wireless Networks*. 2018, pp. 1–8.

[28] B. B. Klebanov, J. Burstein, N. Madnani, A. Faulkner, and J. Tetreault, "Building subjectivity lexicon(s) from scratch for essay data," in *Computational Linguistics and Intelligent Text Processing*, A. Gelbukh, Ed. Berlin, Germany: Springer, 2012, pp. 591–602.

[29] C. Stab and I. Gurevych. (Oct. 2014). *Identifying Argumentative Discourse Structures in Persuasive Essays*. [Online]. Available: http://www.aclweb.org/anthology/D14-1006

[30] B. B. Klebanov, J. Burstein, and N. Madnani, "Sentiment profiles of multiword expressions in test-taker essays: The case of noun-noun compounds," *ACM Trans. Speech Lang. Process.*, vol. 10, no. 3, Jul. 2013, Art. no. 12.

[31] N. Farra, S. Somasundaran, and J. Burstein, "Scoring persuasive essays using opinions and their targets," in *Proc. 10th Workshop Innov. NLP Building Educ. Appl.*, 2015, pp. 64–74.

[32] C. Jiang, F. Coenen, R. Sanderson, and M. Zito, "Text classification using graph mining-based feature extraction," in *Research and Development in Intelligent Systems XXVI*, M. Bramer, R. Ellis, and M. Petridis, Eds. London, U.K.: Springer, 2010, pp. 21–34.

[33] A. B. Massé, G. Chicoisne, Y. Gargouri, S. Harnad, O. Picard, and O. Marcotte, "How is meaning grounded in dictionary definitions?" in *Proc. 3rd Textgraphs Workshop Graph-Based Algorithms Natural Lang. Process. (TextGraphs-3)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 17–24. [Online]. Available: http://dl.acm.org/citation.cfm?id=1627328.1627331

[34] M. Gamon, "Graph-based text representation for novelty detection," in *Proc. 1st Workshop Graph Based Methods Natural Lang. Process. (TextGraphs)*, Jul. 2006, pp. 17–24.

[35] V. S. S. Pillutla, "Helping users learn about social processes while learning from users: Developing a positive feedback in social computing," Northern Illinois Univ., DeKalb, IL, USA, Tech. Rep., 2017.
organization name: organization location: DeKalb, IL, USA, ILLINOIS and report no: Thesis, no report number specified

[36] W. Jin and R. K. Srihari, "Graph-based text representation and knowledge discovery," in *Proc. ACM Symp. Appl. Comput.*, Jan. 2007, pp. 807–811.

[37] M. Litvak and M. Last, "Graph-based keyword extraction for single-document summarization," in *Proc. Workshop Multi-Source Multilingual Inf. Extraction Summarization*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2008, pp. 17–24.

[38] P. Giabbanelli and A. Tawfik, "Overcoming the PBL assessment challenge: Design and development of the incremental thesaurus for assessing causal maps (ITACM)," in *Technology, Knowledge and Learning*. Sep. 2017.

[39] V. K. Gupta, P. J. Giabbanelli, and A. A. Tawfik, "An online environment to compare students' and expert solutions to ill-structured problems," in *Learning and Collaboration Technologies. Learning and Teaching*, P. Zaphiris and A. Ioannou, Eds. Cham, Switzerland: Springer, 2018, pp. 286–307.

[40] P. J. Giabbanelli, A. A. Tawfik, and V. K. Gupta, "Learning analytics to support teachers' assessment of problem solving: A novel application for machine learning and graph algorithms," in *Utilizing Learning Analytics to Support Study Success*. Cham, Switzerland: Springer, 2019, pp. 175–199.

[41] E. Loper and S. Bird, "NLTK: The Natural Language Toolkit," in *Proc. ACL-02 Workshop Effective Tools Methodol. Teach. Natural Lang. Process. Comput. Linguistics (ETMTNLP)*, vol. 1. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002, pp. 63–70. doi: 10.3115/1118108.1118117.

[42] D. S. McNamara, S. A. Crossley, and P. M. McCarthy, "Linguistic features of writing quality," *Written Commun.*, vol. 27, no. 1, pp. 57–86, 2010.

[43] J. Burstein, C. Leacock, and R. Swartz, "Automated evaluation of essays and short answers," Tech. Rep., 2001.

[44] D. W. Reynolds, "Repetition in nonnative speaker writing: More than quantity," *Stud. Second Lang. Acquisition*, vol. 17, no. 2, pp. 185–209, Jun. 1995.

[45] S. Darus and K. Subramaniam, "Error analysis of the written english essays of secondary school students in Malaysia: A case study," *Eur. J. Social Sci.*, vol. 8, no. 3, pp. 483–495, 2009.

[46] K. Allan, "A note on the source of there in existential sentences," *Found. Lang.*, vol. 7, no. 1, pp. 1–18, Feb. 1971.

[47] T. N. Huckin and L. H. Pesante, "Existential there," *Written Commun.*, vol. 5, no. 3, pp. 368–391, 1988.

[48] N. Taguchi, W. Crawford, and D. Z. Wetzel, "What linguistic features are indicative of writing quality? A case of argumentative essays in a college composition program," *Tesol Quart.*, vol. 47, no. 2, pp. 420–430, 2013.

[49] M. T. Gentner, "The impact of coordinating conjunction use on the sentence development of thai and Khmer University student writers," *Panyapiwat J.*, vol. 8, no. 3, pp. 178–187, 2016.

[50] S. Stotsky, "The vocabulary of essay writing: Can it be taught?" *College Composition Commun.*, vol. 32, no. 3, pp. 317–326, 1981.

[51] L. S. Norton, "Essay-writing: What really counts?" *Higher Educ.*, vol. 20, no. 4, pp. 411–442, Dec. 1990. doi: 10.1007/BF00136221.

[52] C. I. Chase, "Essay test scores and reading difficulty," *J. Educ. Meas.*, vol. 20, no. 3, pp. 293–297, 1983.

[53] S. H. Stapa and M. M. Izahar, "Analysis of errors in subject-verb agreement among Malaysian ESL learners," *3L: Lang., Linguistics, Literature*, vol. 16, no. 1, pp. 1–18, 2010.

[54] E. Miltsakaki and K. Kukich, "Automated evaluation of coherence in student essays," in *Proc. LREC*, 2000, pp. 1–8.

[55] Y. Chen, B. Perozzi, R. Al-Rfou, and S. Skiena, "The expressive power of word embeddings," *arXiv:1301.3226*, 2013. [Online]. Available: https://arxiv.org/abs/1301.3226

[56] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.zip: Compressing text classification models," *arXiv:1612.03651*, 2016. [Online]. Available: https://arxiv.org/abs/1612.03651

[57] A. Patel, A. Sands, C. Callison-Burch, and M. Apidianaki, "Magnitude: A fast, efficient universal vector embedding utility package," in *Proc. Conf. Empirical Methods Natural Lang. Process., Syst. Demonstrations*, 2018, pp. 120–126.

[58] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50. [Online]. Available: http://is.muni.cz/publication/884893/en

[59] A. Pawar, S. Budhiraja, D. Kivi, and V. Mago, "Are we on the same learning curve: Visualization of semantic similarity of course objectives," *arXiv:1804.06339*, 2018. [Online]. Available: https://arxiv.org/abs/1804.06339

[60] S. S. Budhiraja and V. Mago, "Extracting learning outcomes using machine learning and white space analysis," in *Proc. 4th EAI Int. Conf. Smart Objects technol. Social Good*, 2018, pp. 7–12.

[61] S. Somasundaran, B. Riordan, B. Gyawali, and S.-Y. Yoon, "Evaluating argumentative and narrative essays using graphs," in *Proc. 26th Int. Conf. Comput. Linguistics (COLING)*, 2016, pp. 1568–1578.

[62] E. D. Kolaczyk and G. Csárdi, "Descriptive analysis of network graph characteristics," in *Statistical Analysis of Network Data With R*. New York, NY, USA: Springer, 2014, pp. 43–67.

[63] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Ann. Hist. Comput.*, vol. 7, no. 1, pp. 43–57, 1985.

[64] E. Jones, T. Oliphant, and P. Peterson. (2001). *SciPy: Open Source Scientific Tools for Python*. [Online]. Available: http://www.scipy.org/

[65] P. Hage and F. Harary, "Eccentricity and centrality in networks," *Social Netw.*, vol. 17, no. 1, pp. 57–63, Jan. 1995.

[66] O. Medelyan, "Computing lexical chains with graph clustering," in *Proc. ACL Student Res. Workshop*, 2007, pp. 85–90.

[67] P. Phandi, K. M. A. Chai, and H. T. Ng, "Flexible domain adaptation for automated essay scoring using correlated linear regression," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 431–439.

[68] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proc. 1st Instructional Conf. Mach. Learn.*, Piscataway, NJ, USA, vol. 242, Dec. 2003, pp. 133–142.

[69] A. M. Mostafa, "An evaluation of sentiment analysis and classification algorithms for arabic textual data," *Int. J. Comput. Appl.*, vol. 158, no. 3, pp. 1–8, 2017.

[70] M. D. Shermis and J. Burstein, *Handbook of Automated Essay Evaluation: Current Applications and New Directions*. Evanston, IL, USA: Routledge, 2013.

[71] J. Burstein, B. Beigman-Klebanov, N. Madnani, and A. Faulkner, "17 automated sentiment analysis for essay evaluation," in *Handbook of Automated Essay Evaluation: Current Applications and New Directions*. 2013, p. 281.

[72] K. Ahmad, Ed., *Affective Computing and Sentiment Analysis: Emotion, Metaphor and Terminology*, vol. 45. Marrakech, Morocco: Springer, 2011.

[73] F. N. Ribeiro, M. Araújo, P. Gonçalves, M. A. Gonçalves, and F. Benevenuto, "SentiBench—A benchmark comparison of state-of-the-practice sentiment analysis methods," *EPJ Data Sci.*, vol. 5, no. 1, p. 23, 2016.

[74] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. 8th Int. AAAI Conf. weblogs Social Media*, 2014, pp. 1–10.

[75] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical Turk: A new source of inexpensive, yet high-quality, data?" *Perspect. Psychol. Sci.*, vol. 6, no. 1, pp. 3–5, Jan. 2011. doi: 10.1177/1745691610393980.

[76] K. J. Berry, J. E. Johnston, and P. W. Mielke, Jr, "Weighted kappa for multiple raters," *Perceptual Motor skills*, vol. 107, no. 3, pp. 837–848, 2008.

[77] J. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," *Pattern Recognit.*, vol. 42, no. 3, pp. 409–424, 2009.

[78] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[79] A. Gulli and S. Pal, *Deep Learning With Keras*. Birmingham, U.K.: Packt Publishing Ltd, 2017.

[80] T. E. Oliphant, *A Guide to NumPy*, vol. 1. Spanish Fork, UT, USA: Trelgol Publishing, 2006.

[81] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[82] S. Boughorbel, J.-P. Tarel, and N. Boujemaa, "Conditionally positive definite kernels for SVM based image recognition," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2005, pp. 113–116.

[83] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[84] H. Chen, B. He, T. Luo, and B. Li, "A ranked-based learning approach to automated essay scoring," in *Proc. 2nd Int. Conf. Cloud Green Comput.*, Nov. 2012, pp. 448–455.

[85] Y. Tay, M. C. Phan, L. A. Tuan, and S. C. Hui, "SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[86] M. Chen and X. Li, "Relevance-based automated essay scoring via hierarchical recurrent model," in *Proc. Int. Conf. Asian Lang. Process. (IALP)*, Nov. 2018, pp. 378–383.

[87] C. Jin, B. He, K. Hui, and L. Sun, "TDNN: A two-stage deep neural network for prompt-independent automated essay scoring," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics* , vol. 1. 2018, pp. 1088–1097.

[88] J. Kumar, R. Goomer, and A. K. Singh, "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters," *Procedia Comput. Sci.*, vol. 125, pp. 676–682, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917328557

[89] P. Coirier and C. Golder, "Writing argumentative text: A developmental study of the acquisition of supporting structures," *Eur. J. Psychol. Edu.*, vol. 8, no. 2, pp. 169–181, Jun. 1993, doi: 10.1007/BF03173160.

[90] N. Ong, D. Litman, and A. Brusilovsky, "Ontology-based argument mining and automatic essay scoring," in *Proc. 1st Workshop Argumentation Mining*, 2014, pp. 24–28.

[91] E. A. Lavin, P. J. Giabbanelli, A. T. Stefanik, S. A. Gray, and R. Arlinghaus, "Should we simulate mental models to assess whether they agree?" in *Proc. Annu. Simulation Symp. (ANSS)*, San Diego, CA, USA: Society for Computer Simulation International, 2018, pp. 6:1–6:12. [Online]. Available: http://dl.acm.org/citation.cfm?id=3213032.3213038

HARNEET KAUR JANDA received the bachelor's degree in information technology from YCCE, Nagpur, India. She is currently pursuing the degree in artificial intelligence and natural language processing in computer science with Lakehead University, ON, Canada. She has two years of work experience as an Application Developer at the Bank of New York Mellon, Pune, India. She is also a Graduate Assistant with Lakehead University. Her research interests include machine learning and natural language processing.

ATISH PAWAR received the B.E. degree (Hons.) in computer science and engineering from the Walchand Institute of Technology, India, in 2014, and the master's degree (Hons.) in computer science from Lakehead University, in 2018. He was with Infosys Technologies, from 2014 to 2016. He has served as a Research Assistant with DataLab, Lakehead University, during the master's degree, where he is currently a Programmer with the Engineering Department. His research interests include machine learning and natural language processing.

SHAN DU received the Ph.D. degree in electrical and computer engineering from The University of British Columbia, Vancouver, BC, Canada, in 2009. She is currently an Assistant Professor with the Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada. Before joining Lakehead University, she was a Research Scientist with IntelliView Technologies, Inc., Canada. She has more than 15 years of research and development experience on image/video processing, image/video analytics, pattern recognition, computer vision, and machine learning. She was a recipient of many awards and grants, including NSERC-IRDF, NSERC-CGS D, AITF Industry Research and Development. Associates Grant, and ICASSP Best Paper Award. She is also a Senior Member of the IEEE Signal Processing society and the IEEE Circuits and Systems Society. She is also an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology, an Area Chair of ICIP 2019, a TPC Member, and a Reviewer for many international journals and conferences.

VIJAY MAGO received the Ph.D. degree in computer science from Panjab University, India, in 2010. He is currently an Associate Professor with the Department of Computer Science, Lakehead University, ON, Canada, where he teaches and conducts research in areas, including big data analytics, machine learning, natural language processing, artificial intelligence, medical decision making, and the Bayesian intelligence. In 2011, he joined the Modelling of Complex Social Systems Program at the IRMACS Centre, Simon Fraser University. He has published extensively (more than 50 peer reviewed articles) on new methodologies based on soft computing and artificial intelligent techniques to tackle complex systemic problems, such as homelessness, obesity, and crime. He has served on the program committees of many international conferences and workshops. In 2017, he joined Technical Investment Strategy Advisory Committee Meeting for Compute Ontario. He currently serves as an Associate Editor for the IEEE Access and *BMC Medical Informatics and Decision Making* and a Co-Editor for the *Journal of Intelligent Systems*.