

AUTOMATED ESSAY SCORING SYSTEM

Amogh Varsh Raju¹, Anugam Sai Kiran², Asmath Fathima³

¹Department of Computer Science and Engineering, SR University, Warangal, Telangana, India.

²Department of Computer Science and Engineering, SR University, Warangal, Telangana, India.

³Department of Electronics & Communication Engineering SR University, Warangal, Telangana,

India.

4

*Email: varshrajuambati@gmail.com

*Email:saikirananugam@gmail.com

Abstract: A big part of evaluating student performance in the educational system is assessment. The current evaluation system uses human evaluation. The manual evaluation procedure becomes more challenging as the number of teachers to students slowly rises. Manual evaluation has a number of shortcomings, including the fact that it takes a lot of time and is unreliable. This online assessment system for connections was developed as a replacement for paper-and-pencil testing procedures. There is currently no adequate system for grading essays or short responses; the computer-based evaluation system only works for multiple-choice questions. For the past few decades, a lot of researchers have been working on automated essay grading and short answer scoring, but it has been difficult to evaluate essays by taking into account all the criteria, such as the content's relevance to the prompt, the development of ideas, cohesion, and coherence. While many scholars addressed style-based assessment, few researchers concentrated on content-based evaluation. A thorough literature assessment of automated essay scoring systems is presented in this essay. We examined the limits of the most recent studies and research trends while studying the Artificial Intelligence and Machine Learning approaches used to assess computerized essay grading. We noticed that the cohesion and relevancy of the essay's content are not taken into consideration when evaluating it.

Keywords Assessment · Short answer scoring · Essay grading · Natural language processing · Deep learning

1. INTRODUCTION

The COVID 19 outbreak has made the need for an online educational system imperative. Almost all educational institutions, from schools to colleges, have adopted the online education system in the current environment. When determining a student's capacity for learning, the assessment is crucial. The majority of automated evaluation is accessible for multiple-choice questions, however it is still difficult to evaluate short and essay replies. The educational system is transitioning to an online environment, including computer-based testing and automated evaluation. It is a critical application that uses machine learning and natural language processing (NLP) methods in the education sector. With simple programming languages and basic methods like pattern matching and language processing, it is impossible to evaluate essays. We will receive more student comments with various explanations in this case, when the issue is with a single question. Consequently, we must assess each response to the question.

A computer-based evaluation method called automated essay scoring (AES) automatically scores or marks student replies by taking into account pertinent criteria. The Project Essay Grader (PEG) by Ajay et al. served as the foundation for the AES research in 1966. (1973). To score the essay, PEG considers writing traits including grammar, diction, composition, etc. The PEG of Shermis et al. (2001) has been improved to focus on grammar checking with a correlation between human and machine evaluators.

Foltz et al. (1999) developed the Intelligent Essay Assessor (IEA) by scoring the substance of an essay using latent semantic analysis. These systems, such as the Bayesian Essay Test Scoring System (BESTY) by Rudner and Liang (2002) and Intellimetric by Rudner et al. (2006), use natural language processing (NLP) methods that concentrate on style and content to determine an essay's grade. Powers et al. (2002) proposed the E-rater. In the 1990s, the vast majority of essay scoring systems used time-honored strategies like pattern matching and statistical analysis. The essay grading systems have started utilising regression-based and natural language processing approaches over the past ten years. Deep learning approaches were applied by AES systems developed in 2014, including those by Dong et al. (2017) and others.

AES systems are used in school instruction in Ohio, Utah, and the majority of US states. For example, Ohio's standardised test (a modernised form of PEG) evaluates responses from millions of students each year. These systems provide feedback to students on their essays and are effective for both formative and summative assessments. Basic essay evaluation criteria were established by Utah and included the following six elements: development of ideas, organisation, style, word choice, sentence fluency, and conventions. For more than a decade, Educational Testing Service (ETS) has been doing extensive research on AES. As a result, ETS has developed an algorithm to evaluate essays on various topics and give test-takers the chance to develop their writing abilities. Additionally, these are content-based evaluations based on recent research.

2 Research method

We framed the research questions with PICOC criteria.

Population (P) Student essays and answers evaluation systems.

Intervention (I) evaluation techniques, data sets, features extraction methods.

Comparison (C) Comparison of various approaches and results.

Outcomes (O) Estimate the accuracy of AES systems,

Context (C) NA.

2.1 Research questions:

To collect and provide research evidence from the available studies in the domain of auto-mated essay grading, we framed the following research questions (RQ):

RQ.1: What are the types of AES methodologies available on field?

This question gives information on types of datasets being used to use certain kind of model only.

RQ.2: What are the features extracted for the assesment of the essay?

The answer to the question can provide insight into various features so far extracted, and libraries used to extract those features.

RQ.3: What are the challenges faced on training the model?

This question gives the information on how the training is being done.

2.2 Search process

On well-known computer science sources as ACL, ACM, IEEE Explore, Springer, and Science Direct, we ran an automated search for an SLR. As much of the research during these years concentrated on cutting-edge technologies like deep learning and natural language processing for automated essay grading systems, we resorted to articles published from 2010 to 2020. Additionally, study in this area was prompted by the availability of open data sets as those from Kaggle (2012) and the Cambridge Learner Corpus-First Certificate in English test (CLC-FCE) by Yannakoudakis et al. (2011).

Search Terms: We conducted a metadata search using search terms such as "Automated essay grading," "Automated essay scoring," "short answer scoring systems," "essay scoring systems," and "auto- matic essay evaluation."

2.3 Selection criteria

We created selection criteria to determine which documents should be included and which should be excluded after gathering all pertinent documents from the repositories. It is easier to conduct reliable and focused research when there are inclusion and exclusion criteria.

inclusion standards 1 We use datasets made up of English-language articles in our analysis. The pieces published in other languages were not included.

For the review, we only considered studies that used the AI technique and eliminated those that used more conventional approaches.

Inclusion criterion 3 Because the study is about essay grading methods, we only included studies that used text data sets, rather than those that used image or speech datasets.

Exclusion standards We eliminated the review papers and survey papers. Also the state of the art papers.

3 . Results

AES System	Approach	Data-set	Features applied	Evaluation Metrics & results
Pedro Uria Rodriguez et al. (2019)	BERT, Xlnet	ASAP Kaggle	Error correction.	QWK 0.755
Jiawei Liu et al. (2019)	CNN, LSTM, BERT	ASAP Kaggle	semanticdata, handcrafted features like grammar correction,essay	QWK 0.709

			length etc	
Darwish and Mohamed (2020)	Multiple Linear Regression	ASAP Kaggle	Style and content-based features	QWK 0.77
Jiaqi Lun et al. (2020)	BERT	SemEval-2013	Student Answer, R	Accuracy 0.8277 (2-way)
Süzen, Neslihan, et al. (2020)	Text mining	Introduction to computer science in UNT, Assignments	Sentence similarity	Correlation score 0.81
Wilson Zhu and Yu Sun in(2020)	RNN (LSTM, Bi-LSTM)	ASAP Kaggle	Word embedding, grammar count, word coun	QWK 0.70
Salim Yafet et al. (2019)	XGBoost machine learning	ASAP Kaggle	Word Count,POS, parse tree,coherence,cohesion,type token ration	Accuracy 68.12
Andrzej Cader (2020)	Deep Neural Network	University of Social Sciences in	asynchronous feature	Accuracy 0.99
Tashu TM, Horváth T (2019)	Rule based algorithm, Similarity based algorithm	ASAP Kaggle	Similarity based feature applied	Accuracy 0.68
Masaki Uto(B) and Masashi Okano (2020)	Item Response Theory Models	ASAP Kaggl		ASAP Kaggl

	(CNN- LSTM,BERT)			
--	---------------------	--	--	--

Total amount of papers: 10

Quality papers 7/10

Maximum data set used: ASAP kaggle

3. ANSWERS:

RQ1: What are the types of AES methodologies available on field?

Answer:

After carefully examining each document, we divide the methods employed by automated essay grading systems into four groups. 1. Regression-based methods. Model for classification. 3. Neural systems. 4. An ontology-based strategy

Supervised learning approaches are used by all of the AES systems that have been created in the last ten years. The AES system was seen by researchers employing supervised methods as either a regression or classification task. The regression task's objective is to forecast an essay's grade. The categorization job entails grouping the essays into those that are (low, medium, or highly) pertinent to the topic of the inquiry. The majority of AES systems created over the past three years have utilized the neural network idea.

RQ:2 :What are the features extracted for the assesment of the essay?

Answer:

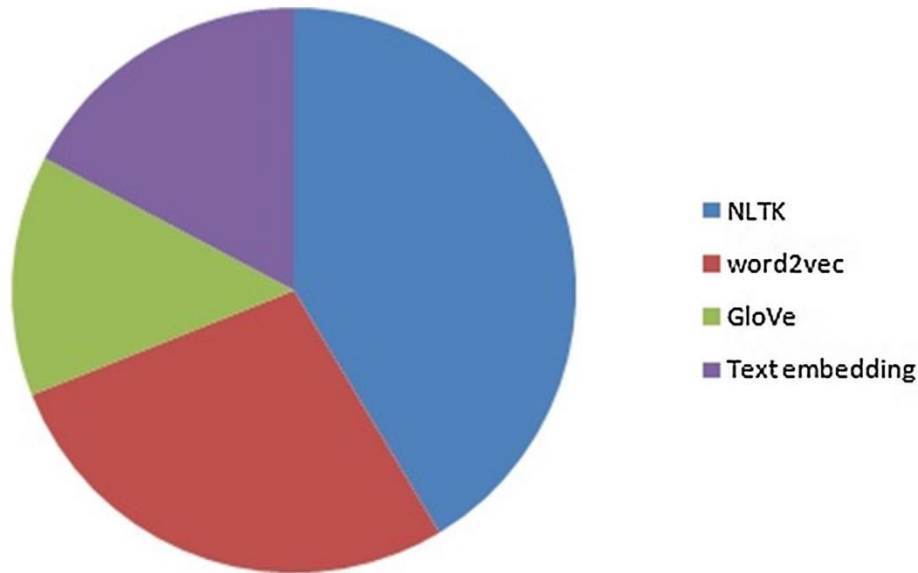
Features play a major role in the neural network and other supervised Machine Learning approaches. The automatic essay grading systems scores student essays based on

Table 4: Types of features

<i>Stastical features</i>	<i>Style based features</i>	<i>Content based features</i>
<i>Essay length with respect to number of words</i>	<i>Sentence structure</i>	<i>Cohesion between senteces in a document</i>
<i>Essay length with respect to sentence</i>	<i>POS</i>	<i>Overlapping(prompt)</i>
<i>Average sentence length</i>	<i>Punctuation</i>	<i>Relevance of information</i>
<i>Average word length</i>	<i>Grammatical</i>	<i>Semantic role of words</i>
<i>N-gram</i>	<i>Logical operations</i>	<i>Correctness</i>
	<i>Vocabulary</i>	<i>Consistency</i>
		<i>Sentence expressing key concepts</i>

ifferent types of features, which play a prominent role in training the models. Based on their syntax and semantics and they are categorized into three groups. 1. statistical- based features Contreras et al. (2018); Kumar et al. (2019); Mathias and Bhattachar- yya (2018a; b) 2. Style-based (Syntax) features Cummins et al. (2016); Darwish and Mohamed (2020); Ke et al. (2019). 3. Content-based features Dong et al. (2017). A good set of features appropriate models evolved better AES systems. The vast majority of the researchers are using regression models if features are statistical-based. For Neu- ral Networks models, researches are using both style-based and content-based features. The following table shows the list of various features used in existing AES Systems. Table 4 represents all set of features used for essay grading.

We studied all the feature extracting NLP libraries as shown in Fig. 3. that are used in the papers. The NLTK is an NLP tool used to retrieve statistical features like POS, word count, sentence count, etc. With NLTK, we can miss the essay's semantic features. To find semantic features Word2Vec Mikolov et al. (2013), GloVe Jeffrey Pennington et al. (2014)



is the most used libraries to retrieve the semantic text from the essays. And in some systems, they directly trained the model with word embeddings to find the score. From Fig. 4 as observed that non-content-based feature extraction is higher than content-based.

RQ.3:What are the challenges faced on training the model?

From our research and the results described in the previous section, many researchers have worked on automated essay scoring systems using numerous techniques. There are statistical methods, classification methods, and neural network approaches for automatically grading essays. The main purpose of automated essay grading systems is to reduce human effort and improve consistency.

The majority of essay scoring systems are concerned with algorithmic efficiency. However, automatic essay grading systems have many challenges. Essays should be evaluated on parameters such as content relevance to the prompt, idea generation, cohesion, coherence, and domain knowledge.

Neither model works for content relevance. That is, if the student's response or explanation is relevant to the prompt given, and if it is relevant to an appropriate degree, there is no debate about the cohesion and coherence of the essay. All research focused on extracting features, training models, and testing results using several NLP libraries. However, there is no discussion of consistency and completeness in essay grading systems, but Palma and Atkinson (2018) describe grading essays based on consistency. Zupanc and Bosnic (2014) also used the term coherence to assess essays. and found coherence using latent semantic analysis (LSA) to find coherence from essays, but the lexicographic meaning of coherence is "logical and coherent disposition".

Another limitation is the lack of assessment of the essay's domain knowledge base using machine learning models. For example, the meaning of cell can vary from biology to physics. Many machine learning models use WordVec and GloVec to extract features. These NLP libraries cannot convert words with more than one meaning to vectors.

3.2.1: Other challenges that influencec the Automates Essay scoring ststems.

All of these approaches worked to improve the model's QWK score. However, QWK does not evaluate the model in terms of feature extraction and produces irrelevant answers. QWK does not evaluate the model, regardless of whether the model evaluates the answer correctly. Student response to automated grading systems presents many challenges. As with the valuation approach, no model has been constructed that considers the valuation method of

Authors	Cohesion	Coherence	Completeness	Feedback
Pedro Uria Rodriguez et al. (2019)	Medium	Medium	Medium	Low
Jiawei Liu et al. (2019)	High	High	Medium	Low
Darwish and Mohamed (2020)	Medium	Low	Low	Low
Jiaqi Lun et al. (2020)	High	High	Low	Low
Süzen, Neslihan, et al. (2020)	High	High	Low	Low
Wilson Zhu and Yu Sun in (2020)	Medium	Medium	Low	Low
Wilson Zhu and Yu Sun (2020)	Medium	Medium	Low	Low
Salim Yafet et al. (2019)	High	Medium	Low	Low
Tashu (2020)	Medium	Medium	Low	Low
Tashu and Horváth (2020)	Medium	Medium	Low	Medium

Table: Comparision of all approaches on various features.

Approach es	Gram mar	Style (Word choice, senten ce struct ure)	Mechan ics (Spellin g,punc-	Developm ent	BoW (tf-idf)	relevan ce
Pedro Uria Rodriguez et al. (2019)	No	No	No	No	Yes	Yes
Jiawei Liu et al. (2019)	Yes	No	No	Yes	No	Yes
Darwish and Mohamed (2020)	Yes	Yes	No	No	No	Yes
Jiaqi Lun et al. (2020)	No	No	No	No	Yes	Yes
Süzen, Neslihan, et al. (2020)	Yes	No	No	Yes	No	Yes
Wilson Zhu and Yu Sun in (2020)	Yes	Yes	No	Yes	Yes	Yes
Jiawei Liu et al. (2019)	Yes	No	No	Yes	No	Yes
Salim Yafet et al. (2019)	No	No	No	No	No	Yes
Tashu (2020)	Yes	No	No	Yes	No	Yes
Tashu and Horváth (2020)	No	No	No	No	No	Yes

4. Syntesis

In our systematic literature search on automated essay grading systems, we first collected 25 papers containing selected keywords from various databases. After inclusion and exclusion criteria, 15 articles remained. To these selected papers, we applied the quality criteria of the two peer reviewers and finally selected 10 papers for final peer review.

Here are our observations on the automatic grading system for essays from 2018 to 2020:

- *Implementation techniques for automated essay scoring systems can be divided into four areas. 1. Regression models 2. Classification models 3. Neural networks*

4. Ontology-based methods, but researchers using neural networks are more accurate than others, and all methods are state-of-the-art as shown in Table 3. Thing.

- *Most of the regression and classification models for essay scoring used statistical features to determine the final score. This means that algorithms, parameters extracted from essays, are not directly trained on essays, but systems or models are trained on parameters such as word count, sentence count, etc. Algorithms trained on some numbers taken from the essay get good scores when the numbers match the configuration. Otherwise the rating will be lower. In these models, the evaluation process is based solely on numbers, regardless of the essay. Therefore, using statistical parameters to train the algorithm will likely lead to inconsistent and relevant essays.*

- *In the neural network approach, the model is trained with a Bag of Words (BoW) function. The BoW feature has no word-to-word relationship between the semantic meaning of words and sentences. Example: Set 1:*

John killed Bob. Set 2:

Bob killed John. In these two sentences, the BoW is "John", "killed", and "bob".

- *In the Word2Vec library, when you unidirectionally create a word vector from an essay, that vector has dependencies on other words and finds semantic relationships with other words. But if a word has more than one meaning, as in "bank loan" and "river bank," he said that "listen to the bank" has two meanings, and the adjacent words determine the meaning of the sentence. increase. In this case Word2Vec cannot find the true meaning of the word from the sentence.*

- *In the AES system, consistency is the main characteristic to consider when evaluating essays. The true meaning of consistency is togetherness. It is the logical connection of sentences (local level consistency) and paragraphs (global level consistency) in a story. Without consistency, every sentence in a paragraph stands alone and has no meaning. In essays, consistency is a key feature that describes all of the flow and its meaning. Finding essay semantics is a powerful feature of the AES system. Consistency allows you to assess whether all sentences are connected in flow and whether all paragraphs are connected to justify the prompt. Obtaining a level of consistency from essays is an important task for all researchers in AES systems.*
- *While we're talking about voice ratings, the data set contains up to 1 minute of audio. Feature extraction techniques are fundamentally different from text scoring and vary in accuracy with fluency, pitch, male to female, boy to adult voices. However, the training algorithm is the same for text scoring and audio scoring.*
- *Given that AES systems accurately evaluate essays and short answers in all directions, there is a great demand for automated systems in the educational and related worlds. The AES system is now used on his GRE and TOEFL exams. Apart from these, Coursera (["https://coursera.org/learn/machine-learning/exam"](https://coursera.org/learn/machine-learning/exam)) and NPTEL (<https://swayam.gov.in/explorer>) also offer multiple choice Assess student performance with questions. From another perspective, the AES system can be used in information retrieval systems such as Quora, Stack Overflow, etc. to check whether the retrieved answers are suitable for the question and to rank the retrieved answers. .*

5.Data set

We have used OS data-set for this project, the data set consists of 3 independent variables and 2 dependent feature they are as follows.

Response: Whole essay (sentences format)

Review 1: Score from first review

Review 2: Score from second review

Word choice: Score for type of words picked and vocab.

Organization: Score for the organization of the sentences

		that contain a computer ??? from cellular phones and video gam			
A	B	C	D	E	F
iD	Response	Reviewer-1	Reviewer-2	word choice	Organization
1	An operating system (OS) is system	4	4	3	1
1	An operating system is the most im	5	5	2	3
1	Collection of programs that manage	2	1	1	1
1	It is an interface user and machine(I	2	1	1	0
1	An operating system is a software w	3	2	2	1
1	It is a platform for humans to intera	1	1	1	1
1	An operating system (OS) is system	5	5	3	3
1	software which act as interface betw	3	2	2	1
1	Operating System is a software syst	4	4	2	1
1	An operating system (OS) is system	4	4	2	2
1	Operating system is nothing but a so	2	2	2	1
1	An operating system, or OS is softw	2	2	1	1
1	It is the interface between compute	2	2	1	1
1	An operating system (OS) is system	3	3	1	1

6.Pre-processing

We have used a transformer to complete this process where the main goal is to convert the data into vector, perform embedding on it and tokenise it.

*The data set consists of 2391 * 6 rows and columns respectively. These are the steps taken for data Pre-processing.*

Word Embedding:

Word embedding is a method for translating words to vectors of real numbers in language modelling. It represents words or sentences in a multidimensional vector space. Numerous techniques, including neural networks, co-occurrence matrices, probabilistic models, etc., can be used to create word embedding. Word embedding generation models are part of Word2Vec. With one input layer, one hidden layer, and one output layer, these models are shallow two-layer neural networks. Word2Vec uses two different architectures.

Tokenization:

Tokenization is the process of dividing text into a list of tokens from a string of text. Tokens can be viewed as components, similar to how a word functions as a token in a phrase and how a sentence functions as a token in a paragraph:

- *Text into sentences tokenization*
- *Sentences into words tokenization*
- *Sentences using regular expressions tokenization*

The initial stage in any NLP pipeline is tokenization. It significantly affects the remainder of your pipeline. Tokenization is the process of dividing unstructured data and natural language text into

units of data that can be regarded as discrete pieces. A document's token occurrences can be directly used as a vector to represent that document. This instantly converts a text document or unstructured string into a numerical data format appropriate for machine learning. They can also be directly employed by a computer to initiate helpful answers and actions. Alternatively, they could be employed as features in a machine learning pipeline to initiate more complicated actions or behaviour.

7. Methodology

1. Transfer learning:

The Vanishing Gradient problem, which impairs long-term memory, affected RNN. RNN processes text in a sequential manner, therefore if a sentence is long like "XYZ visited France in 2019 during a time when there were no cases of cholera," it will be processed as "XYZ visited France in 2019 during a time when there were no cases of cholera." Now, if we inquire as to which location is meant by "that country" here? The fact that the country was "France" won't be remembered by RNN because it has already heard the term "France" many times. The model was trained at the word level, not at the sentence level, due to the sequential nature of processing. When the gradient shrinks, no true learning occurs because the gradients convey information used in the RNN parameter update.

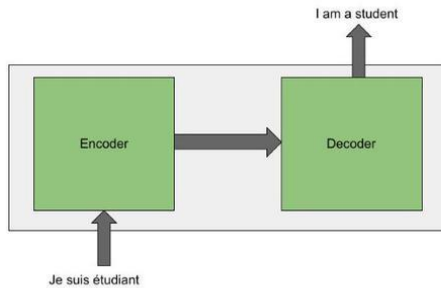


Fig.7.1 Architecture

The issue of long-term memory loss was partially handled by adding a few extra memory cells and addressing the vanishing gradients problem. However, because RNN was unable to process the entire sentence at once, the issue with sequential processing persisted. It processed words sequentially as opposed to in concurrently. Due to their sequential architecture, LSTMs cannot resolve this problem. We employ the static embedding strategy in LSTMs, which proposes that we embed a word into an n -dimensional vector without beforehand understanding its context. However, the meaning also changes if the context does.

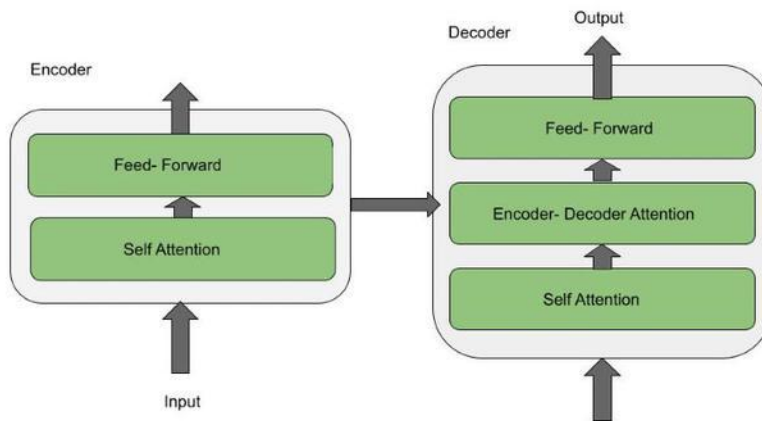


Fig. 7.2 Encoder and decoder

Self Attention and Feed Forward are the two layers of the encoder architecture. The outputs of the self-attention layer are supplied to a feed-forward neural network once the encoder's inputs have passed through it. Sequential data possesses temporal properties. It means that each word has a certain place in relation to the others. Take the line, "The cat didn't chase the mouse because it was not hungry," as an example. Here, it is clear that "it" refers to the cat, but it is more difficult to understand for an algorithm. Self-attention enables the model to connect the word "it" with the word "cat" when it is processing the word "it". Self-attention is the approach to reformulate the representation depending on all other words of the sentence.

The Self Attention, Encoder-Decoder Attention, and Feed Forward layers make up the decoder architecture. In addition to the self-attention and feed-forward layers found in the encoder, the decoder also features an attention layer that aids in focusing on key elements of the input sentence.

In the Transformer architecture, there are six layers of encoders and decoders. Word embeddings are carried out at the bottom encoder, where each word is converted into a 512-byte vector. The output of the encoder directly below would serve as the input to the other encoders. The encoder's many levels are used to identify the NLP pipeline. As an example, part of speech

tags are employed in the first layer, constituents in the second, dependencies in the third, semantic roles in the fourth, coreference in the fifth, and relations in the sixth.

The very last layer, known as Softmax, assigns a probability to each word in the lexicon, and all of these probabilities add up to 1.

Code snippet:

```
[ ] from sentence_transformers import SentenceTransformer

[ ] model = SentenceTransformer('sentence-transformers/all-mpnet-base-v2')
    embeddings = model.encode(d)
```

Fig : 7.3

```
[ ] from sentence_transformers import SentenceTransformer, util
    model = SentenceTransformer('multi-qa-MiniLM-L6-cos-v1')

    query_embedding = model.encode('An operating system (OS) is system software th')
    passage_embedding = model.encode(['An operating system is the most important s',
                                     'An operating system is a software which acts a.'])

    print("Similarity:", util.dot_score(query_embedding, passage_embedding))
```

Downloading: 100%	<div></div>	737/737 [00:00<00:00, 13.5kB/s]
Downloading: 100%	<div></div>	190/190 [00:00<00:00, 5.22kB/s]
Downloading: 100%	<div></div>	11.5k/11.5k [00:00<00:00, 160kB/s]
Downloading: 100%	<div></div>	612/612 [00:00<00:00, 16.8kB/s]
Downloading: 100%	<div></div>	116/116 [00:00<00:00, 1.71kB/s]
Downloading: 100%	<div></div>	25.5k/25.5k [00:00<00:00, 336kB/s]
Downloading: 100%	<div></div>	90.9M/90.9M [00:02<00:00, 29.3MB/s]
Downloading: 100%	<div></div>	53.0/53.0 [00:00<00:00, 927B/s]
Downloading: 100%	<div></div>	112/112 [00:00<00:00, 1.12kB/s]
Downloading: 100%	<div></div>	466k/466k [00:00<00:00, 528kB/s]
Downloading: 100%	<div></div>	383/383 [00:00<00:00, 653B/s]
Downloading: 100%	<div></div>	13.8k/13.8k [00:00<00:00, 2.09kB/s]

The screenshot shows a Google Colab notebook with the following content:

```
clustered_sentences = [[] for i in range(num_clusters)]
for sentence_id, cluster_id in enumerate(cluster_assignment):
    clustered_sentences[cluster_id].append(X[reviewer-2])

for i, cluster in enumerate(clustered_sentences):
    print('Cluster: ', i+1)
    print(cluster)
    print("")
```

The output displays four clusters of sentences, each identified by a reviewer ID (e.g., 2389, 2385, 2386, 2387, 2388, 2389) and a cluster ID (e.g., 2, 3, 1, 4). The sentences are grouped into these clusters based on their similarity.

Fig:7.4 : Result snippets

2. K-Means Clustering

(Imagining objects as points in an n -dimensional space will assist.) The items will be divided up into k groups or clusters of resemblance by the algorithm. We will use the euclidean distance as a unit of measurement to calculate that similarity.

This is how the algorithm operates:

First, we randomly initialise k locations, also known as cluster centroids or means.

Each item is categorised according to the nearest mean, and the coordinates of that mean, which are the averages of the items categorised in that cluster thus far, are updated.

After a specific amount of iterations, we repeat the process until we get our clusters.

Since the things described in the "points" mentioned above have mean values, they are known as means. We can initialise these means in a variety of ways. Initializing the means at random elements in the data set is a natural approach. Another approach is to put the means' beginning

values at arbitrary ranges between the data set's boundaries (if for a feature x the items have values in $[0,3]$, we will initialise the means with values for x at $[0,3]$).

Classify Items:

Now we need to create a function to group or cluster an item. We will compare the similarity of the provided item to each mean before classifying it with the closest one.

Find means:

We will loop through every item to classify them into the closest cluster and update the cluster's mean before actually determining the means. The procedure will be repeated a certain number of times. If no item's classification changes during the course of two rounds, the procedure is terminated because the algorithm has found the best course of action.

The function below accepts as inputs the items, the maximum number of iterations, and k (the number of desired clusters), and outputs the means and the clusters. An item's classification is kept in the array `belongs To`, while the size of a cluster is kept in `cluster-sizes`.

Find clusters:

Finally, given the means, we wish to identify the clusters. Each item will be categorized into the nearest cluster when we have gone through all of the objects.

The other widely used similarity metrics include:

1. Cosine distance: It establishes the angle's cosine between two locations in an n -dimensional space using the formula $d = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$

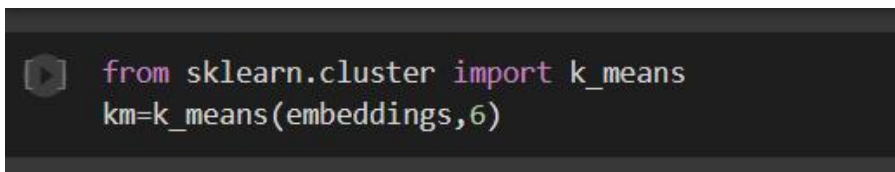
$$\frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

2. *Manhattan distance: The total of the absolute differences between the coordinates of the two data points is computed.*

$$d = \sum_i |X_i - Y_i|$$

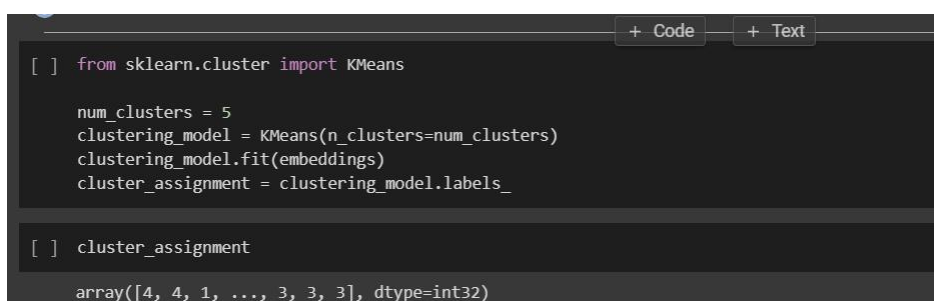
3. *The generalised distance metric is another name for the Minkowski distance. Both ordinal and quantitative variables may be used with it.*

$$d = \left(\sum_i |X_i - Y_i|^p \right)^{\frac{1}{p}}$$

A dark-themed code editor window showing two lines of Python code. The first line imports k_means from sklearn.cluster. The second line creates a KMeans model with 6 clusters and fits it to an embeddings array.

```
from sklearn.cluster import k_means
km=k_means(embeddings,6)
```

Fig.7.5

A dark-themed code editor window showing Python code for KMeans clustering. The code imports KMeans, sets the number of clusters to 5, creates a KMeans model, fits it to embeddings, and prints the cluster assignment. The output shows an array of cluster labels: [4, 4, 1, ..., 3, 3, 3].

```
[ ] from sklearn.cluster import KMeans

num_clusters = 5
clustering_model = KMeans(n_clusters=num_clusters)
clustering_model.fit(embeddings)
cluster_assignment = clustering_model.labels_

[ ] cluster_assignment

array([4, 4, 1, ..., 3, 3, 3], dtype=int32)
```

Fig 7.6

8.Results

8.1: Kappa score:

```
result=cohen_kappa_score(lis,abc,weights='quadratic')
print(result)

[ ]
... 0.09624577645852728
```

8.2: Embeddings

```
from scipy.cluster import hierarchy
threshold = 0.1
Z = hierarchy.linkage(embeddings,"average", metric="cosine")
C = hierarchy.fcluster(Z, threshold, criterion="distance")
print(embeddings,Z,C)

[[-0.02357353 -0.01503747 -0.00397871 ... 0.02411804 0.0362772
 0.0074688 ]
[-0.02846667 0.03639808 0.00822516 ... 0.03154779 0.01749238
-0.00048567]
[-0.00468063 -0.02878353 -0.03404774 ... 0.00117501 -0.02464179
0.00338477]
...
[-0.02753562 -0.01181827 -0.02199969 ... -0.03585221 0.05486577
-0.01896431]
[ 0.01241688 -0.06888344 -0.03817156 ... -0.02222495 -0.00727084
-0.03246032]
[-0.01234783 -0.00325741 -0.02912418 ... -0.04279251 0.02785357
-0.00177822]] [[0.00000000e+00 6.00000000e+00 0.00000000e+00 2.00000000e+00]
[2.59000000e+02 2.85000000e+02 0.00000000e+00 2.00000000e+00]
[9.00000000e+00 2.39000000e+03 0.00000000e+00 3.00000000e+00]
...
[4.73000000e+03 4.74900000e+03 8.64300949e-01 4.00000000e+00]
[4.77400000e+03 4.77600000e+03 9.09938693e-01 2.36600000e+03]
[4.77500000e+03 4.77700000e+03 9.64334096e-01 2.39000000e+03]] [736 734 831 ... 209 264 206]
```

```

clustered_sentences = [[] for i in range(num_clusters)]
for sentence_id, cluster_id in enumerate(cluster_assignment):
    clustered_sentences[cluster_id].append(X['Reviewer-1'])

for i, cluster in enumerate(clustered_sentences):
    print("Cluster ", i+1)
    print(cluster)
    print("")

```

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

Streaming output truncated to the last 5000 lines.

```

2386    3
2387    2
2388    1
2389    2
Name: Reviewer-1, Length: 2390, dtype: int64, 0      4
1      5
2      2
3      2
4      3
..
2385    2
2386    3
2387    2
2388    1
2389    2
Name: Reviewer-1, Length: 2390, dtype: int64, 0      4

```

8.3: clustered data

7.4:

```

Name: Reviewer-1, Length: 2390, dtype: int64, 0      4
1      5
2      2
3      2
4      3
..
2385    2
2386    3
2387    2
...
2388    1
2389    2
Name: Reviewer-1, Length: 2390, dtype: int64]

```


9. Conclusion

Essays are collections of sentences and paragraphs that are useful to analyze the writing, communication, and grammatical skills of users or applicants. Essays became a standard evaluation criterion in several fields like secondary education, academics, software recruitment's etc. As there are huge number of applicants or participants, it's a hurdle for human evaluators to assess each essay and score it. It will kill huge amount of time and delay the process.

We have created a new way of analyzing the essay and scoring them based on clustering which helps the data to be easily be classifies into categories and the kind of scoring being offered will be easily understood based on the group of answers and their similarity.

The clustering through K-Means clustering helps the categorization but the sentence transformer takes the embedding very seriously and helps in easy access of the data to to be converted into vectors and helps in tokenization too.

This model has a deep understanding of sentences and the similarities between the sentences, hence more useful to create vectors with perfect meaning. We are trying to reduce the burden of essay evaluators and make the work automated. This project will doesn't show any bias in generating the score.

10. References

1. Sharma A., & Jayagopi D. B. (2018). Automated Grading of Handwritten Essays 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), 2018, pp 279–284. <https://doi.org/10.1109/ICFHR-2018.2018.00056>
2. Agung Putri Ratna, A., Lalita Luhurkinanti, D., Ibrahim I., Husna D., Dewi Purnamasari P. (2018). Auto- matic Essay Grading System for Japanese Language Examination Using Winnowing Algorithm, 2018 International Seminar on Application for Technology of Information and Communication, 2018, pp. 565–569. <https://doi.org/10.1109/ISEMANTIC.2018.8549789>.
3. Wu, S. H., & Shih, W. F. (2018, July). A short answer grading system in chinese by support vector approach. In Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications (pp. 125-129).
4. Kumar, N., & Dey, L. (2013, November). Automatic Quality Assessment of documents with application to essay grading. In 2013 12th Mexican International Conference on Artificial Intelligence (pp. 216– 222). IEEE.
5. Zhu W, Sun Y (2020) Automated essay scoring system using multi-model Machine Learning, david c. wyld et al. (eds): mlnlp, bdiot, itccma, csity, dtmn, aifz, sigpro
6. Tashu TM, Horváth T (2020) Semantic-Based Feedback Recommendation for Automatic Essay Evaluation. In: Bi Y, Bhatia R, Kapoor S (eds) Intelligent Systems and Applications. IntelliSys 2019. Advances in Intelligent Systems and Computing, vol 1038. Springer, Cham
7. Tashu TM, Horváth T (2019) A layered approach to automatic essay evaluation using word-embedding. In: McLaren B, Reilly R, Zvacek S, Uhomoibhi J (eds) Computer Supported Education. CSEDU 2018. Communications in Computer and Information Science, vol 1022. Springer, Cham
8. Tashu TM (2020) "Off-Topic Essay Detection Using C-BGRU Siamese. In: 2020 IEEE 14th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, p 221–225, doi: <https://doi.org/10.1109/ICSC.2020.00046>
9. Rodriguez P, Jafari A, Ormerod CM (2019) Language models and Automated Essay Scoring. ArXiv, abs/1909.09482
10. Parekh S, et al (2020) My Teacher Thinks the World Is Flat! Interpreting Automatic Essay Scoring Mechanism.” ArXiv abs/2012.13872 (2020): n. pag
Others:
- Cai C (2019) Automatic essay scoring with recurrent neural network. In: Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications (2019): n. pag.
- Ke, Z., Inamdar, H., Lin, H., & Ng, V. (2019, July). Give me more feedback II: Annotating thesis strength and related attributes in student essays. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 3994-4004).
- Kumar, Y., Aggarwal, S., Mahata, D., Shah, R. R., Kumaraguru, P., & Zimmermann, R. (2019, July). Get it scored using autosas—an automated system for scoring short answers. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 9662–9669).
- Liang G, On B, Jeong D, Kim H, Choi G (2018) Automated essay scoring: a siamese bidirectional LSTM neural network architecture. Symmetry 10:682

Liu J, Xu Y, Zhao L (2019) Automated Essay Scoring based on Two-Stage Learning. ArXiv, abs/1901.07744

Lun J, Zhu J, Tang Y, Yang M (2020) Multiple data augmentation strategies for improving performance on automatic short answer scoring. In: Proceedings of the AAAI Conference on Artificial Intelligence, 34(09): 13389-13396

Muangkammuen P, Fukumoto F (2020) Multi-task Learning for Automated Essay Scoring with Senti-ment Analysis. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop p 116–123

Parekh S, et al (2020) My Teacher Thinks the World Is Flat! Interpreting Automatic Essay Scoring Mechanism.” ArXiv abs/2012.13872 (2020): n. pag

Riordan B, Flor M, Pugh R (2019) "How to account for misspellings: Quantifying the benefit of character representations in neural content scoring models."In: Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications

Rodriguez P, Jafari A, Ormerod CM (2019) Language models and Automated Essay Scoring. ArXiv, abs/1909.09482

Rudner, L. M., & Liang, T. (2002). Automated essay scoring using Bayes' theorem. The Journal of Technology, Learning and Assessment, 1(2).

Süzen, N., Gorban, A. N., Levesley, J., & Mirkes, E. M. (2020). Automatic short answer grading and feedback using text mining methods. Procedia Computer Science, 169, 726–743.

Xia L, Liu J, Zhang Z (2019) Automatic Essay Scoring Model Based on Two-Layer Bi-directional Long- Short Term Memory Network. In: Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence p 133–137