Automated Marking System forEssay Questions

Abstract
The stress of marking assessment scripts of many candidates often results in fatigue that could lead to low productivity and reduced consistency. In most cases, candidates use words, phrases and sentences that are synonyms or related in meaning to those stated in the marking scheme, however, examinersrely solely on the exact words specified in the marking scheme. This often leads to inconsistent grading and in most cases, candidates are disadvantaged. This study seeks to address these inconsistencies during assessment by evaluating the marked answer scripts and the marking scheme of Introduction to File Processing (CSC 221)from the Department of Computer Science, University of Uyo, Nigeria.These were collected and used with the Microsoft Research Paraphrase (MSRP) corpus. After preprocessing the datasets, they were subjected to Logistic Regression (LR), a machine learning technique where the semantic similarity of the answers of the candidates was measured in relation to the marking scheme of the examiner using the MSRP corpus model earlier trained on the Term Frequency-Inverse Document Frequency(TF-IDF) vectorization. Results of the experiment show a strong correlation coefficient of 0.89and a Mean Relative Error (MRE) of 0.59 compared with the scores awarded by the human marker (examiner). Analysis of the error indicates that block marks were assigned to answers in the marking scheme while the automated marking system breaks the block marks into chunks based on phrases both in the marking scheme and the candidates' answers. It also shows that some semantically related words were ignored by the examiner.
Keywords: MSRP corpus, Semantic similarity, Machine Learning, Logistic Regression, Marking Scheme, TF-IDF, Natural Language Processing.

1.0 Introduction

In marking essay examination papers, a student's answer is compared with the answer in the marking scheme of the examiner. The scheme is normally structured in words, clauses, phrases and sentences. The marks awarded to a student are expressed as the degree of the relatedness of the student's answer to the examiner's marking scheme. An automated marking systemdoes not require students to provide the exact lexical structure of the marking scheme but rather the semantic similarity and sometimes a paraphrase of the marking scheme is considered.

Semantic similarity refers to the semblance between two entities such as words, clauses, phrases and sentences in terms of their meanings while lexical similarity refers to the semblance of two concepts especially words in terms of the structure of the co-occurring words.

The coexistence of many possible meanings for a word or phrase being nearly or the same in meaning to another word or phrase is a challenge in the field of Natural Language Processing (NLP). Computational models'development is one way of addressing issues in NLPto resolve the challenge of semantic relations between concepts. Relations such as hierarchical ('is-a') associative (cause part), and 'part of' are studied in NLP and used to develop models that compute the degree of similarity between concepts. The hierarchical relation in particular is used to view the classification of concepts (taxonomy) according to their similarities and differences. The degree of similarity of words can also be computed using a large collection of words or texts through which a queried word could be searched. Mostly, the semantic

similarity of words could be computed either through the corpus-based approach or through a knowledge-based approach.
With electronic learning gaining currency, there is a need for an automated marking that will provide a platform for assessing electronic essay-based submissions.Currently assessing the Multiple-Choice Questions (MCQ) does present so many challenges. MCQ only deals with the exactness of an answer or word and does not require human-cognitive reasoning ability of approximation and fuzziness (Obot et al., 2023). In the conventional learning system, both the MCQ and essay questions are required to assess and test student's understanding and knowledge of a subject. To achieve success with the adoption of e-Learning in Higher Education assessment and evaluating the learner's understanding, there is a need to incorporate an evaluation system where theoretical questions are marked and graded electronically.

Identification of such paraphrases needs a corpus of paraphrases as provided in the MSRP corpus. The MSRP is widely used in the paraphrase recognition/identification task, being the baseline to compare different algorithms. The corpus contains 5801 pairs of sentences/phrases which have been extracted from news sources on the web, along with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship (Alvi et al., 2021). The WordNet, a lexical database of words, senses and their semantic relations is used to find the synonyms and hypernyms of words using the ontology of synsetthat can also be used to perform a similar task.

During assessments,students are at liberty to use words that are similar or related to what is in the marking scheme to express their knowledge and understanding. Most times, what is expected from a student is an expression of an idea that is similar to that expressed in the marking scheme. This idea expressed by the candidate is a paraphrase of the examiner's expectation.

This study addresses the imperative need for an Automated Marking System that not only expedites the marking process but also enhances accuracy and provides timely and consistent assessment feedback. The objectives include the collection and processing of pertinent datasets, the integration of advanced machine-learning techniques, and a comprehensive evaluation of the system's performance. In Section 2, the literature on key concepts is reviewed and presented while in Section 3, the materials and methods of the study are presented. Results and discussion of the results are presented in Section 4and theconclusion of the research is presented in Section 5.

2.0 Literature Review

The similarity of words, phrases, and sentences in a document could be based on lexical structure or the degree of semantic relatedness of a pair of words, phrases or sentences in a document. The semantic relatedness can be measured based on knowledge-based or corpus-based. There are several corporals used in this regard, including Microsoft Research Paraphrase (MSRP),(Alvi et al., 2021), Clough and Stevenson (CS) and Webis Crowd Paraphrase Corpus 2011 corpora(Vrbanec and Meštrovi´c (2020)).

Yang et al (2019) developed semantic document classification based on strategies of semantics similarity computations and correlation analysis. They identified the problems of polysemy and synonyms as the main issues that cause misclassification in documents and

proposed a novel method of strong correlation analysis. The polysemy problem was resolved usingthe semantic similarity computing (SSC) method. This involves a text document being split into sentences and the contents from the dictionary of each word in the sentence are extracted based on its part-of-speech tag in the sentence. This is semantically compared with each concept of a word with thesentence and returns the concept with the maximum similarity score. Words that cannot be used to determine their exact meanings are excluded from the list while those with more distinct terms are selected. The category discrimination method (CDM) followed by establishing a correlation between the word and other feature words and measuring the categories by the feature correlation analysis (FCA) was used to resolve the synonym problem. TF-IDF was used to implement at a preprocessing phase.

Two sets of data, one from a collection of 18758 vocabularies taken from Rotten Tomatoes and another from 56821 Chinese news were used for the experiment where a baseline Convolutional Neural Networks(CNN) was taken as an example to compare the performance of classical Neural Networks (NN) and the improved one with the proposed strategy in document classification. The results obtained from the improved model were compared with those obtained using classical machine learning classifiers and found to covary positively with a slight positive accuracy in favour of the improved method.

Kholodna et al (2022) developed the detection of paraphrases by the binary of text pairs using various NLP tools such as the Jaccard coefficient, Cosine distance, word mover distance, wordnet etc. Unified modelling language such as use case diagram, activity diagram and class diagram were also employed in the design. Machine learning tools like Siamese NN based on recurrent NN were applied. The system development principle is based on stacking with an NN of 2 hidden layers of 512 and 128 neurons and an output layer of 16 neurons. A logistic regression classifier was used to reduce computing resources. Implementation of the model was done usingthePython programming language. Results obtained show accuracy on a test dataset of 92.46%, area under curve ROC = 97.05%, area under precision-recall curve = 94.96% while accuracy on validation datasets of 91.71%, area under ROC = 97.66% and area under precision-recall curve = 96.12% were also recorded.

Onyshchenko et al (2022) identified the similarities of two sentences through paraphrasing by considering the BERT base, RoBERTa base and ALBERT-based models using the MSRP corpus for training and testing in Siamese, triplet neural networks and various versions of logistic regression. Each of the models was trained for 30 epochs with cosine as the similarity measure. Comparing the results obtained for triplet networks and logistic regression, the use of the neural network-based measure of similarity showed much better results, especially when using the RoBERTa and ALBERT models. A combined approach that uses the BERT-like models for fine-tuning showed significant improvements in the results.

Synonymous substitution, word reordering, and insertion/deletion have been identified by Alvi et al (2021) as some of the common paraphrasing strategies used by plagiarists. A method to identify synonymous substitution and word reordering in paraphrased plagiarized sentence pairs was therefore proposed. Context matching and pre-trained word embeddings were used to identify synonymous substitution and word reordering. The input data consists of pairs of source and paraphrased sentences available as the Subcorpus of Paraphrased

Sentences extracted from the Corpus of Plagiarised Short Answers. Smith-Waterman Algorithm for Plagiarism Detection and ConceptNet Numberbatch pre-trained word embeddings produced the best performance in terms of F1 scores.

Enikuomehin and Dosumu (2017), proposed an improved Levenshtein distance between two strings (question and answer). The improved model uses the triangular inequality to identify the relationship between the two terms to measure the similarity of terms, and then an Application Programming Interface (API) assisted semantic matching for a subjective online examination system was built. The concepts of text summarization, term dependencies, semantic tagging and corporal buildup were employed in the development of the API. Results show self-grading of essay-type questions using a web-based semantic API.

Motivated by the fact that two sentences may be similar without having identical words Abdalgader and Skabar(2010) proposed a new sentence similarity measure that used word sense disambiguation and synonym expansion to provide a richer semantic context to measure sentence similarity. Results of implementation show a better performance than those found inprevious studies.

Udoh et al (2022) subjected 5025-course materials to retrieval processes using fuzzy logic, Dice, Cosine, Okapi and Jaccard similarity measures models.The average of 3 human experts' scores was used as the base measure. Results of the comparison showed the fuzzy logic model to covary very strongly with the base measure than the results of the other models.

To aid in the retrieval of similar land dispute cases for easy and fast administration of court cases, Obot et al (2023) applied Cosine, Jaccard, Text Semantic Similarity (TSS) and fuzzy logic separately to 205 cases. Results show cosine similarity measure had the strongest correlation, (72%) followed by Jaccard (70%), fuzzy logic (70%) and TSS (63%). They recommended the integration of cosine with fuzzy logic for the design of a decision support system for land dispute case retrieval systems.

Vrbanec and Meštrovi´c (2020) used LSI, TF-IDF, Word2Vec, Doc2Vec, GloVe, FastText, ELMO, and USE to measure the semantic similarity of texts on MSRP Corpus, Clough and Stevenson and Webis Crowd Paraphrase Corpus 2011 corpora. Text pre-processing scenarios, hyper-parameters, sub-model selection, distance measures, and semantic similarity/paraphrase detection threshold were varied. Evaluation of the models was done in terms of accuracy, precision, recall, and F1 measure on three corpora. Results of the experiments conducted reveal that the best thresholds and standard evaluation measure values from training datasets are quite diverse between different models in the same corpus.

Mahmoud and Zrigui (2020) noticed that conventional similarity measures such as TF-IDF, GloVE, and Word2Vec cannot capture efficiently hidden semantic relations when sentences may not contain any common words, or the co-occurrence of words is rarely present. Therefore, they proposed a deep learning model based on Global Word embedding (GloVe) and Recurrent Convolutional Neural Network (RCNN). A paraphrased corpus preserving both semantic and syntactic features of Arabic sentences was developed with their original words replaced by their synonyms with the same POS from a vocabulary. With different

topologies of paraphrase constructed, the results of experiments carried out revealed that the new GloVe-RCNN model based on recurrent structure has achieved the highest results compared to the state-of-the-art methods.

Vrbanec and Meštrović (2021), demonstrated the use of MSRP, Webis and CS to train 4 deep learning models and to measure the similarity of sentences using Cosine, Soft cosine, Euclidean and Manhattan measures. Results obtained showed the superiority of Euclidean over others in terms of Accuracy (0.983), Precision (0.937), Recall (0.980) and F1 (0.957). Research reveals that conventional similarity measures may not always determine the perfect matching without a noticeable relation or concept overlap between two measurable sentences. Consequently, an algorithm to solve this problem using corpus-based ontology and grammatical rules was developed. Experiments conducted on the developed algorithm showed a significant performance improvement in sentences and shorttexts with arbitrary syntax and structure. The machine learning-based measure gave the best performance in terms of accuracy, precision, recall and F1 when compared with the lexicon-based and corpus-based measures.

Mohamed and Oussalah (2020) combined a CatVar database enhanced WordNet semantic similarity measure with Wikipedia-named entity semantic relatedness and normalized Google distance to develop a hybrid system to measure the extent to which a pair of words, phrases and sentences are semantically related. The limitations of WordNet were addressed by the CatVar database. The MSRP and TREC-9 question variants corpora were used to validate the developed system. Results were compared with existing supervised and unsupervised systems and found to perform better in accuracy and precision by 7%.

Jaccard, Cosine, Jaro and Dice similarity measures were used in Obot et al (2021) to mark 647 short answers (subjective) to questions based on the answers given by the candidates and the marking guide generated by the examiners. The similarity of the scores obtained through each of the 4 similarity measures and that obtained through 3 examiners was measured. Scores generated by the Jaro measure were found to covary most strongly with the average scores of 3 examiners with a covariance of 97% and variance error of 62% at 0.001 level of significance.

Jaro similarity measure was used in Obot et al (2023) to compute the degree of similarity between the model answers and the student answers of 500 Multiple Choice Questions (MCQ) collected from 2 universities in Nigeria. Results of the experiment show an average deviation of 13.3 marks from those marked manually by the examiners. The results are encouraging but could be improved on with semantic similarity measure hybridized with string similarity.
Ferreira et al (2018) combined sentence similarity measures to identify paraphrases while Nguyen et al (2019) presented a novel method of learning short-text semantic similarity with word embeddings.

Antonius et al (2020) applied the combination of NLP and machine learning techniques to improve the accuracy of plagiarism detection while Ullah (2021) used machine learning techniques to identify software plagiarism in several programming languages. Naïve Bayes is

used inMwaro et al (2020) for resume selection and classification. Wahdan et al (2020) categorized BBC news using random forest(RF), Logistic regression(LR) and K-nearest neighbour (KNN) and found KNN to score 97%, LR = 96% and RF = 94% on the average of the Accuracy, Precision, Recall and F1 performance measures.
Ullah et al (2018) researched to detect software plagiarism by collecting programs written in C, C++, Java, C# and Python programming languages. These were subjected to training and testing using Softmax regression after principal component analysis was done to reduce dimensionality. The classification accuracy for the training datasets was 84% and 73% for the testing datasets.

Chimingyang (2020) used 12776 datasets from Kaggle to train Long and Short-Term Memory (LSTM) networks and multinominal logistic regression models. Word2Vec embeddings and manually crafted parameters were used on the LSTM model while manually crafted parameters were used on the multinominal logistic regression model. Quadratic Weighted Kapper (QWK) and accuracy were used as evaluation parameters and LSTM was found to perform better than the multinominal Logistic model with QWK of 0.94 and accuracy of 0.32.

WordNet, word sense disambiguation and open NLP library were employed by Hazar et al (2019) to develop a system that identifies the grading of students'answers to essay questions and compares the results with the human grading of the same answers. Results were compared with the existing corpus-based such as ISA Wikipedia and ESA Wikipedia, knowledge-based based such as Leacock and Chodrow, Resnik, Jiang and Convath and Baseline such as Tf-Idf. The developed system was found to have a higher correlation coefficient of 0.490 and a least RMSE of 0.63.

The software developed from our study will help examiners of essay questionsand related examinations in grading the answers provided by candidates of such examinations. The grading process is stress-free, devoid of sentiments, bias and inconsistencies usually associatedwith such marking exercises in the traditional marking system.

3.0 **Materials and Method**

The Flow Diagram for the system is presented in Figure 1.

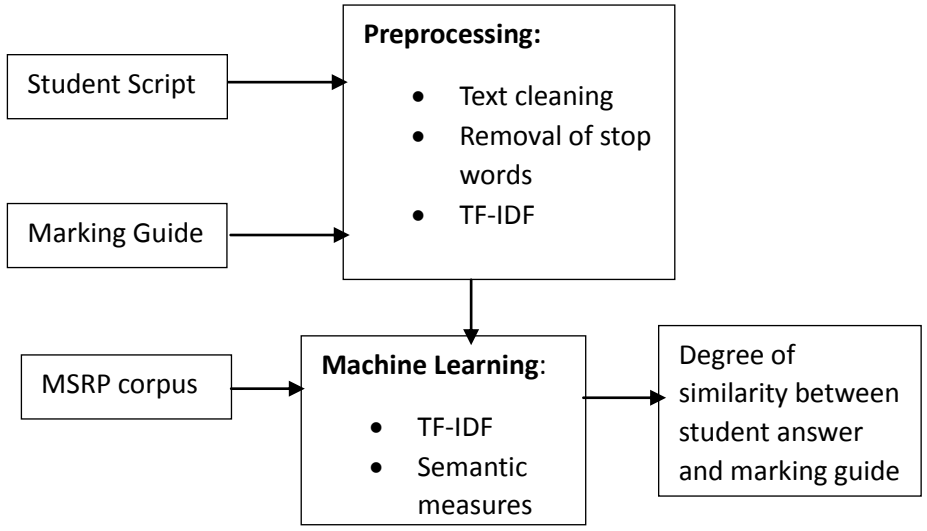

Figure 1: Flow Diagram

3.1 Data Gathering:

This research employs two principal datasetsnamely:

i) University of Uyo Computer Science Department Dataset on CSC 221(Introduction to file processing) which comprises the students' answers to the examination questions and the lecturer's marking guide received in MSWord format transcribed into an Excel format for subsequent analysis.

ii) theMSRParaphraseCorpus from Microsoft Research Community.

University of Uyo Dataset

There were challenges in processing the University of Uyo Dataset, which initially came in MSWord files containing both student answers and marking schemes. The files were carefully transcribed into Ms-Excel to expand the research possibilities.The University of Uyo Computer Science Department Dataset was received as tendered documents in Microsoft Word Format from the Department of Computer Science after they were word processed from the manually written form submitted by the students and marked by the examiners. The first document (DocA) contained entries of 100 students' scripts during the 2019/2020 academic session of course code: CSC 221 (Introduction to File Processing) and the second document (DocB) is the Marking Scheme document created by the lecturers who taught and marked the course.

Table 1 shows how the documents were defined, extracted and processed into meaningful data attributes providing a structured foundation for subsequent analyses.

Table 1: Data Gathering Formulation for Uniuyo Dataset

Sno	Attribute	Meaning	Data Type	ExtractedFrom
1	Qno	The question number attempted by the student, for example, 1,2,3…	String	DocA
2	SubQ	Sub questions of (1) if any, for example, ai, ii,ii, bi, bii	String	DocA
3	StudentRegNo	The registration number of the student	String	DocA
4	Answers	The answers submitted by the student	String	DocA
5	MarkObtainable	The mark obtainable in each question		DocB
6	MarkingScheme	The expected answer provided by the lecturer in a sentence or paragraph	String	DocB
7	ActualMark	The actual score the student was awarded by the lecturer for each	String	DocA

question.

The corresponding values for the attribute names in Table 1 were organized into a Microsoft Excel 2019 Sheet and were saved as MarkingDataSheet.xls. A sample representation of the dataset is presented in Table 2.

Table 2: Sample representation of the Uniuyo Dataset

qNo	Subq	MarkObtainable	StudentRegNo	Answers	MarkingScheme	ActualMark
1a	I	1	CO/28	File processing is the collection of data by creation, updating, merging, etc of files for the computer system to process for an output result or information	File processing is the arrangement or sorting of file structures and organization. It involves updates, maintenance and enquiries.	1
1c	I	2	CO/530	Ginfo66 = (7110511011166)2 5.055936684 x 1025 5.055936684000000000000 Mid value = 0000	71 105 110 102 111 6 6	1
1c	Ii	2	CO/530	Maths 16 = (1019711610411516.)2 1.039811768000000000000000000 Mid value = 0000	109 97 116 104 115 1 6	1

In order to get insights into how the marks in the dataset were distributed across different questions, we used a histogram to visualize the distribution of the 'MarkObtainable' column. Figure 2 represents the distribution of the marks.

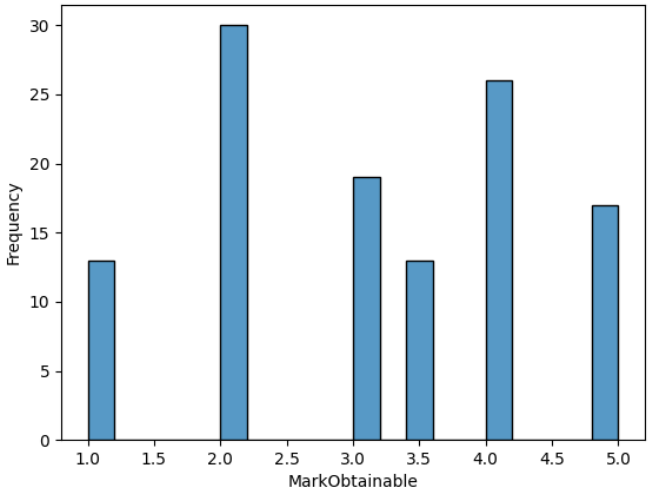

Figure 2: Distribution of MarkObtainable Columns

Figure 2 shows that 30 attempted questions have 2 marks obtainable each and 25 attempted questions have 4 marks each. The precise breakdown of the bar graph in Figure 2 is as follows:

Mark 1.0: The bar for mark 1.0 is short, indicating that there were 13 attempts on questions carrying 1 mark. Students avoided the questions because it has few marks.

Mark 2.0: The bar for mark 2.0 is the tallest, showing that there were 30 attempts approximately on the questions carrying two marks.

Mark 3.0: The bar for mark 3.0 is of medium height, indicating that there were 19 attempts approximately on questions carrying three marks. This could indicate also that the tougher the questions, the higher the mark. This shows also that the questions were not biased.
Mark 3.5: The bar for mark 3.5 is short, similar to mark 1.0, showing there were 13 attempts on questions carrying a 3.5 mark. This could also indicate that much effort is required for questions with higher marks, and it is observable that there were fewer attempts on those questions.
Mark 5.0: The bar for mark 5.0 is short, showing there were 17 attempts on questions carrying 5 marks.
This suggests that the assessment was moderately difficult, with most students attempting questions whose scores were around the middle of the range. Figure 3 shows the Frequency of Questions answered.

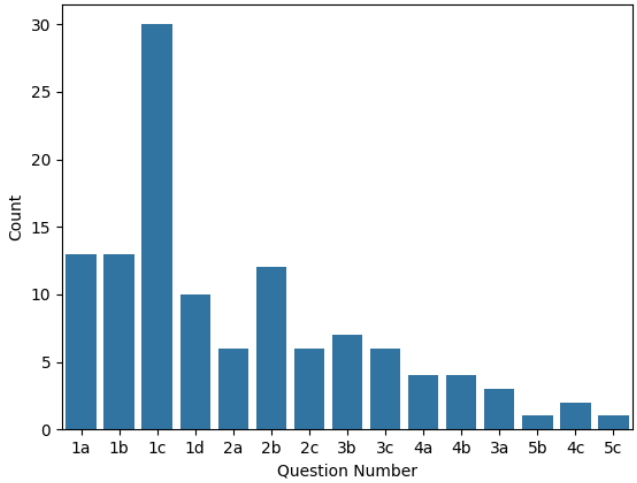

Figure 3: Frequency of Question numbers answered

In Figure 3, question 1c has the highest frequency, most students answered this question because it has sub-questions (i-v). This could be attributed to the fact that questions in 1c could be easy to answer. More insights were found from the Box Plot of the distribution of marks awarded for each question as visualized in Figure 4.

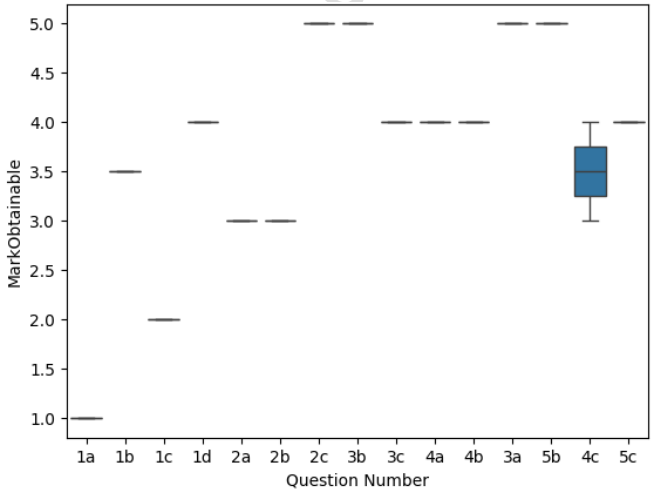

Figure 4: Box Plot showing Marks obtainable by Question

The findings from the Box Plot show the following:

First Quartile (Q1): The bottom of the box is just above 3.0. This is the 25th percentile, meaning that about 25% of students received a score of 3.0 or lower.
Median: The line inside the box is at approximately 3.75. This is the 50th percentile, meaning that about 50% of students received a score of 3.75 or lower.
Third Quartile (Q3): The top of the box is just below 4.5. This is the 75th percentile, meaning that about 75% of students received a score of 4.5 or lower.
Maximum: The upper whisker extends to exactly 5.0. This is the maximum score, indicating that some students achieved full marks.
The other questions (1a to 5c, excluding 5b) have fixed marks obtainable, represented by the horizontal lines at various heights. In our dataset, there were some rows without answers from the students, so Null entries were handled from the University of Uyo datasets ensuring the no NaN values during the calculations.

Feature Extraction

(i) Removal of Stop words

Before the extraction, stop words such as "the", "is", "and"were removed during the text preprocessing step because we considered them to be of little value for our marking task also for fairness, we allowed such common words to be removed for a less strict marking of the student's answers. For strict marking, we recommend the use of stop words.

(ii) Term Frequency-Inverse Document Frequency (TF-IDF)

The transformative power of Term Frequency-Inverse Document Frequency (TF-IDF) was harnessed to extract meaningful features from sentences in the String1, String2 columns of the MSRCorps and the Answers and MarkingScheme columns of the MarkingSheetDataset of the Uniuyo Dataset.

3.2 MSRParaphraseCorpus

The dataset consists of 5801 pairs of sentences/phrases gleaned over 18 months from thousands of news sources on the web. Accompanying each pair is judgment reflecting whether multiple human annotators considered the two sentences to be close enough in meaning to be considered close paraphrases. The MSRParaphraseCorpus was received in text form, a sample of the raw dataset is presented in Figure 5.

#1 String	#2 String	Quality
Amrozi accused his brother, whom he called "the witness", of deliberately distorting his evidence.	Referring to him as only "the witness", Amrozi accused his brother of deliberately distorting his evidence.	1
Yucaipa owned Dominick's before selling the chain to Safeway in 1998 for $2.5 billion.	Yucaipa bought Dominick's in 1995 for $693 million and sold it to Safeway for $1.8 billion in 1998.	0
They had published an advertisement on the Internet on June 10, offering the cargo for sale, he added.	On June 10, the ship's owners had published an advertisement on the Internet, offering the explosives for sale.	1
Around 0335 GMT, Tab shares were up 19 cents, or 4.4%, at A$4.56, having earlier set a record high of A$4.57.	Tab shares jumped 20 cents, or 4.6%, to set a record closing high at A$4.57.	0
The stock rose $2.11, or about 11 percent, to close Friday at $21.51 on the New York Stock Exchange.	PG&E Corp. shares jumped $1.63 or 8 percent to $21.03 on the New York Stock Exchange on Friday.	1
Revenue in the first quarter of the year dropped 15 percent from the same period a year earlier.	With the scandal hanging over Stewart's company, revenue the first quarter of the year dropped 15 percent from the same period a year earlier.	1

Figure 5: MSRParaphraseCorpus

The MSRParaphraseCorpus originally had five attributes namely;

i. Quality
ii. #1 ID
iii. #2 ID
iv. #1 String
v. #2 String

But for the sake of this work, we built the model using three attributes related to our work. Table 3 shows how we labelled the data before training the MSRParaphraseCorpus.

Table 3: Labelled Data from MSRParaphraseCorpus

Sno	Attributes	Label
1	#1 String	Input
2	#2 String	Input
3	Quality	Output

3.3 Machine Learning Algorithm

The machine learning algorithm used for the training of the MSRParaphraseCorpus is Logistic Regression (LR). LRwas chosen as the machine learning algorithm for its suitability for binary classification tasks, interpretability, and efficiency, also from the literature, there has been a previous success in similar tasks, and scalability to handle the dataset's size.

Model Training

The MSRParaphraseCorpus whose features were extracted was split into 70:30 percent portions as inputs for training and testing of the machine learning model respectively. A thorough exploration of the model's architecture, the intricacies of the training process, and the strategic decision behind framing the problem as a regression task are explained here. The trained model was then saved for future use in predicting paraphrase quality.

3. 3.1 Logistic Regression (LR)

To understand the workings of logistics regression, we attempted to mathematically design the process of logistic regression in order to gain insight into the working of the algorithm on our dataset. From our MSRParaphraseCorpus, we selected a row instance, to explain the Logistic Regression.

X_train_sample = ["A bird is flying in the sky.", "An aeroplaneis taking off.", "A man is playing a guitar.", "A woman is riding a bike.", "The sun is shining."]
y_train_sample = [1, 0, 1, 0, 1]

From the Quality column of the dataset, we assumed that a “1” indicates that String 1 is a paraphrase of String 2, and a “0” indicates for non-paraphrase. We represented these sentences using TF-IDF vectors. We used a small vocabulary to aid in the design:
vocabulary = ["a", "bird", "is", "flying", "in", "the", "sky", "an", "aeroplane", "taking", "off", "man", "playing", "guitar", "woman", "riding", "bike", "sun", "shining"]

Using simplified calculation, we compute the TF-IDF Matrix for the paraphrases;
X_train_tfidf_sample = [0.2, 0.4, 0.2, 0.4, 0.3, 0.4, 0.4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

$0, 0, 0, 0, 0, 0, 0, 0.3, 0.5, 0.5, 0.5, 0, 0, 0, 0, 0, 0, 0, 0$
$0.3, 0, 0.3, 0, 0, 0, 0, 0.3, 0, 0, 0, 0.4, 0.4, 0.4, 0, 0, 0, 0, 0$
$0.3, 0, 0.3, 0, 0, 0, 0, 0, 0, 0, 0, 0.3, 0, 0, 0.4, 0.4, 0, 0, 0$
$0, 0, 0, 0, 0, 0.3, 0.3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.4, 0.4, 0.4]$

It should be noted that the number of features in our model corresponds to the number of words in the vocabulary and that the value of each feature Xi is determined by the TF-IDF score for the corresponding word in the sentence being analyzed.

Learning the Weights in LR

Logistic Regression aims to learn the weights (coefficients) for each feature (TF-IDF value) and the bias term. The learned weights are represented as $\boldsymbol{w}$ (weights) and $\boldsymbol{b}$ (bias). For our simplified example, the weights and bias are initialized as follows:

w = $[w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9, w_{10}, w_{11}, w_{12}, w_{13}, w_{14}, w_{15}, w_{16}, w_{17}, w_{18}, w_{19}, w_{20}]$

b = 0.5

Predictions

Logistic Regression makes predictions using the logistic function:

$$z_i = \sigma(w_i \cdot X_i + b_i) \text{for} i = 1,2, \ldots, n) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(1)}$$

where:

$\sigma$is the sigmoid (logistic) function.

$w \cdot X$is the dot product of weights and features.

$b$is the bias term.

In our case, for the first row, we have

$$z = w_1 \cdot 0.2 + w_2 \cdot 0.4 + \cdots + w_{20} \cdot 0 + b \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(2)}$$

After obtaining z, we pass it through the sigmoid activation function. The sigmoid function is defined in Equation 3

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(3)}$$

Let's assume z=0.4 based on the value of a guessed weight then

$$\sigma(z) = \frac{1}{1 + e^{-0.4}} \quad = \ 0.5987$$

So, for the given weights and bias, the predicted probability (p) for the first row is approximately 0.5987.

Gradient Descent

The model was trained using gradient descent to minimize the loss and the gradients of the loss the weights and bias were calculated, and the weights and bias were updated in the opposite direction of the gradients. The gradient descent methods are partial derivatives and are presented mathematically in Equations 1.4 and 1.5 respectively.

$$w \leftarrow w - \alpha \frac{\partial Loss}{\partial w} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(4)}$$

$$b \leftarrow b - \alpha \frac{\partial Loss}{\partial b} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \text{(5)}$$

where:

$\alpha$is the learning rate.

The gradients were calculated using the chain rule of calculus.

Iteration

The steps were iteratively performed until convergence, updating the weights and bias to minimize the loss of the training data. It is important to note that the actual implementation was carried out onPython's sci-kit-learn and this library handles the complexity of these mathematical details, also that the provided explanation is a simplified overview of the training process.

Model Testing:

The remaining 30% of the MSRParaphraseCorpus served as the litmus test for evaluating the model's performance. The choice of Mean Relative Error (MRE)) as an evaluation metric is justified, underlining their relevance in capturing the nuances of the grading system. The Print screen from our program showing training and testing is presented in Figure 6.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

# Load the training data
train_data_path = r'C:\MSRParaphraseCorpus\msr_paraphrase_train.txt'
test_data_path = r'C:\MSRParaphraseCorpus\msr_paraphrase_test.txt'

train_df = pd.read_csv(train_data_path, delimiter='\t', quoting=3)
test_df = pd.read_csv(test_data_path, delimiter='\t', quoting=3)

# Step 1: Data Preprocessing
X_train = train_df[['#1 String', '#2 String']]
y_train = train_df['Quality']

X_test = test_df[['#1 String', '#2 String']]
y_test = test_df['Quality']

# Step 2: Feature Engineering
vectorizer = TfidfVectorizer(stop_words='english', max_features=5000)
X_train_tfidf = vectorizer.fit_transform(X_train['#1 String'] + ' ' + X_train['#2 String'])
X_test_tfidf = vectorizer.transform(X_test['#1 String'] + ' ' + X_test['#2 String'])

# Step 3: Model Training
model = LogisticRegression()
model.fit(X_train_tfidf, y_train)

# Step 4: Model Evaluation
y_pred = model.predict(X_test_tfidf)
accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")
print("\nClassification Report:")
```

Figure 6: Print screen from program showing training and testing

In the program,**model = LogisticRegression()** is used to initialize the constructor of the Logistic Regression Class and saved into the model object**. model.fit(X_train_tfidf, y_train)** is used to call the fit method which is used for the training.

Accuracy of Model

Because the machine learning training task involves building a model that predicts class between binary (0/1) labels 0 – whether the string 2 IS NOT a paraphrase of string 1 or 1

whether string 2 IS a paraphrase of string 1) based on the predicted probabilities obtained from the sigmoid function, we use the accuracy score to measure how many predictions the model got correct out of the total predictions from MSRParaphrase Corpus. The accuracy score is calculated as:

$$accuracy = \frac{number\ \ of\ correct\ predictions}{total\ \ number\ \ of\ predictions}$$

The classification report table below provides a comprehensive overview of the performance metrics for the Logistic Regression model on the 30% test dataset. These metrics include accuracy, precision, recall, and F1-score, providing insights into the model's ability to correctly classify instances of both classes (0 and 1). The table also presents support values, indicating the number of instances in each class. Understanding these metrics is crucial for evaluating the model's effectiveness in distinguishing between paraphrased (class 1) and non-paraphrased (class 0) sentences. The result is shown in Table 4.

Table4: Performance Metrics for LR-model on 30% dataset

Metric	Value
Accuracy	0.69
Precision (0)	0.65
Precision (1)	0.69
Recall (0)	0.16
Recall (1)	0.95
F1-score (0)	0.26
F1-score (1)	0.80
Support (0)	578
Support (1)	1147

4. 0 Results and Discussion

A comprehensive presentation of intermediate results, including MarkAwardedProbability, PredictedMark, and ActualMark, forms the cornerstone of the analysis. The justification for selecting MRE as the primary evaluation metric is explored, emphasizing its effectiveness in regression tasks. After the training and testing of the model trained and tested with a 70-30% split of the MSRParaphraseCorpus, the model and the TF-IDF Vectoriser used for the training were saved for the actual prediction. For simplicity in the reporting of this work, we established the following variables:

Probability of Similarity (PS): This is the probability predicted by the LR that shows how close the answer supplied by the student is to the answer provided by the lecturer.

Mark Obtainable (MO): Every question and sub-question carries a definite mark defined in DOCB.

Actual Mark (AM): This is the actual mark awarded by the expert, in our case, the lecturer who taught and marked the examination or assessment.

Predicted Mark (PM): This is the mark predicted by the Logistic Regression (LR) for each question answered by the student. This is a decimal since it was computed using the probabilities from the LR

$$PM = PS * MO$$

Table5 shows the inputs and the outputs of the prediction of student's marks on each question using the model trained from the MSRParaphrased dataset. For the sake of representing this result, we present only the Answer, Marking Scheme, AM, PS and PM columns of the result.

Table 5: Minimized results showing the Predicted mark of 10 instances.

Answers	MarkingScheme	AM	PS	PM
File processing is the collection of data by creation, updating, merging, etc of files for the computer system to process for an output result or information	File processing is the arrangement or sorting of file structures and organization. It involves updates, maintenance and enquiries.	1	0.824	0.824
45 39 8 54 77 38 24 16 4 7 9 20 | 8 39 45 38 24 16 4 7 9 20 54 77 | 8 39 38 24 16 4 7 9 20 45 54 77 | 8 38 24 16 4 7 7 20 39 45 54 77 | 8 24 16 4 7 9 20 38 39 45 54 77 | 8 16 4 7 9 20 24 38 39 45 54 77 | 8 4 7 9 16 20 24 38 39 45 54 77 | 4 7 8 9 16 20 24 38 39 45 54 77	45 39 8 54 77 38 24 16 4 7 9 20 | 4 39 8 54 77 38 24 16 45 7 9 20 | 4 7 8 54 77 38 24 16 45 39 9 20 | 4 7 8 54 77 38 24 16 45 39 9 20 | 4 7 8 9 77 38 24 16 45 39 54 20 | 4 7 8 9 16 38 24 77 45 39 54 20 | 4 7 8 9 16 20 24 77 45 39 54 38 | 4 7 8 9 16 20 24 38 45 39 54 77 |	3	0.587	2.054
A = 64 J = 73 S = 82 B = 65 K = 74 T = 83 C = 66 L = 75 U = 84 D = 67 M = 76 V = 85 E = 68 N = 76 W = 86 F = 69 O = 78 X = 87 G = 70 P = 79 Y = 88 H = 71 Q = 80 Z = 89 I = 72 R = 81	71 105 110 102 111 6 6	1.4	0.548	1.095
a = 97 j = 106 s = 115 b = 98 k = 107 t = 116 c = 99 l = 108 u = 117 d = 100 m = 109 v = 118 e = 101 n = 110 w = 119 f = 102 o = 111 x = 120 g = 103 p = 112 y = 121 h = 104 q = 113 z = 122	109 97 116 104 115 1 6	1.4	0.679	1.357
	86 111 69 100 117 4 4	1.4	0.545	1.090
	67 79 77 80 115 9 1	1.4	0.699	1.399
	80 72 97 82 77 8 8	1.4	0.801	1.602
The binary search technique is considered to be the most efficient search method.	The binary search involves the continuous division of the blocks into 2 parts ie left to the midpoint and a midpoint to the right. The midpoint is the point where left and right have an equal number of elements on both sides. This method is effective because it is easy to know which side of the point the element will be found where the smaller numbers are found to the left and the larger numbers are found to the right. a1 a2 a3 a4 a5 a6 a7 a8 a9 a10 3 5 9 11 15 17 22 25 37 68 (a) Midpoint is between 15 and 17; when key X is 22	1	0.627	2.507

	3 5 9 11 15 17 22 25 37 68 The key does not match the middle point; it is larger than the value, hence the search moves to the right 22 25 37 68 17 22 25 37 68 The search does not match the point which is 22, so the search moves to the left 15 22 25 Here search matches the number (22). The key (22) is found on block a7			
)	Key = 8 3 5 9 11 15 17 22 25 37 68 The values at the midpoint (15, 17) do not match the key (8), they are larger, so the search moves to the left of the array. 3 5 9 11 15 This midpoint (9) doesn't match the key (8), it is larger so the search moves leftward 3 5 9 * This does not match the key (8), it is a smaller value so 8 cannot be found in the search.	0	0.708	2.832
File Permission i. Read permission: one can view its content ii. Write permission: one can change content. iii. Execute permission. Contents can be run or executed.	The concept of file permission is used to categorize the users' permission. It is important to ensure that the system files are not opened. The UNIX operating system separates access control on files and directories according to its characteristics that is owner, group or other system. Permission: Bits capable of being set or reset to allow certain types of access to it. To determine the permission assigned to the various users is called the Unix operating system. The result from the long format may take the result of the general format. R= Read; W = Write, X = Execute Comand: % LS-C = Long format Listing	2	0.769	2.308

During the prediction, some factors were considered.

i. Answer columns: Answer columns were converted to a string and Empty answer columns with NaN were converted to empty string.

ii. Students who supplied empty answers: To avoid NaN error for probabilities, since blank spaces were not removed in the answer sentences when we used TF-IDF, we manually used a controlled structure to set the values of PS and PM to 0. In subsequent work, the use of language models that consider blank spaces can be investigated to further improvement of the system.

iii. Computing the Predicted Mark: Since LR returns the probabilities [0,1] that an answer submitted by a student is similar to the marking scheme and to what extent, so to compute the predicted mark for each question, we multiplied the MOP for that row by the MO as shown in Table 5.

A line plot showing the comparison of the actual mark and the predicted mark by the LR is presented in Figure 6.

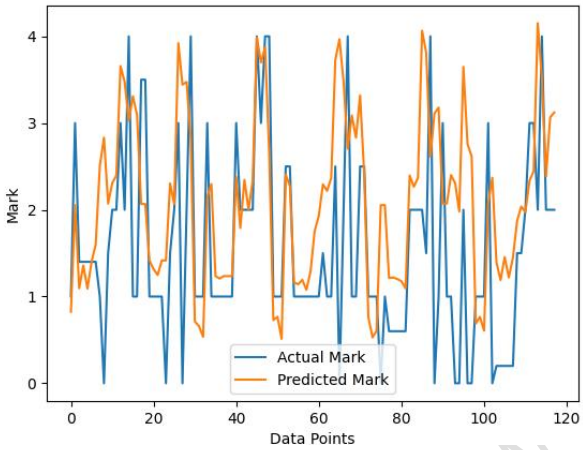

Figure 7: Line Plot showing Actual vs Predicted Mark

A detailed comparison between the total predicted marks generated by the Automated Marking System and the actual total marks awarded by lecturers for 9 students is presented. Table 6 shows the mark awarded toeach student identified by a registration number.

Table 6: Experiment Results: Comparison of Predicted and Actual Total Marks

Index	StudentRegNo	Total Predicted Mark	Total Actual Mark
1	CO/28	32.227061	28.5
2	CO/495	11.476508	18.0
3	PH/1326	9.714845	12.0
4	CO/495	4.712337	3.5
5	CO/530	29.699378	38.0
6	CO/542	40.021354	34.0
7	CO/590	30.996982	23.5
8	CO/687	16.337345	7.0
9	CO/698	33.044211	30.0

In Table7, the indices, Student Registration Numbers, and the corresponding marks provide insights into the system's performance across various student submissions. Deviations between predicted and actual marks are evident, emphasizing the importance of further system refinement. The disparities stemmed from diverse factors such as the complexity of marking schemes which includes marking or marking or even unmarked questions due to human error, variations in subjective interpretation, and challenges in capturing nuanced aspects of student answers such as diagrams and tables. These results serve as a foundation for future enhancements and discussions on the efficacy of the Automated Marking System.

Performance Evaluation

The Mean Relative Error (MRE) is the metric used to evaluate the performance of regression models.

Table 7: Performance measure for Logistic Regression Model for Predicting Student Mark

Metric	TF-IDF+LR	Word2Vec + LR
Mean Relative Error (MRE)	0.5919	0.5101

Mean Relative Error (MRE): This is the average of the absolute differences between the predicted and actual values, divided by the actual values. It is a measure of the relative size of the errors. Our MRE of 0.5919 means that on average, the model's predictions were off by about 59.19% relative to the actual values.

Likely Causes of Errors

Improper marking of scripts led to incorrectly labelled data:

(i) therewas overmarking in some of the cases, so the actual labels for such instances were improperly labelled.

(ii) In some cases, the marking scheme was not followed as we noticed under marking too, marks were awarded without considering the mark obtainable in the DOCB

(iii) Most questions with -sub-question were lump-summed and awarded marks as a question, meanwhile the dataset was organized to handle each unit of the question as a unique question.

5.0 Conclusion

With the increasing number of students in a class and the corresponding examinations, teachers are confronted with the challenge of accurately assessing their performance in examinations. The stress and strain associated with this exercise often lead to inconsistencies and errors in marking. Inconsistency in marking or scoring where a fair score is awarded to a student who has the same or similar expressed answers as another who is awarded a different score.

In this study, a design concept was set out to solve the problem of inconsistency and sentiments in the scoring of students in an essay examination where candidates use different words, phrases, clauses and sentences to express their answers to questions. These words, phrases, clauses and sentences though different may connote the same or similar meanings to those in the marking scheme of the examiner. Sometimes, the examiner while assessing the script of the student may not be able to decipher the meaning of the words etc and in the process mark down the candidate. The Microsoft Research Paraphrase (MSRP) corpus with more than 5810 English paraphrases has been used in several anti-plagiarism software. In this study, the MSRP corpus was used with a corpus of examination documents (students' answer scripts and examiner's marking schemes) and an inference conducted on these documents results in the score of a candidate based on the maximum score or point allotted for each question in the examiners' marking scheme. A machine learning approach of Logistic Regression was used in the marking exercise after the corporal was subjected to preprocessing tasks of data cleaning, transcribing, stop words removal and TF-IDF. The marking of 100 scripts undertaken shows a strong correlation of 0.8879between predicted

scores with the actual scores awarded by the human markers (examiners). The MRE of 0.5919 was also recorded.

It was noticed that during the marking exercise, the examiners awarded block marks to the students sometimes inconsistent with the marking guide. Also, the marking guide is not comprehensive in the allocation of marks as many sentences may be lumped up and assigned some marks. This posed a challenge to the automated marking exercise in deciding which point superseded the other. These issues coupled with the fact that some words and paraphrases used by the students though similar in meaning to that in the marking scheme were ignored resulting in the student being marked down contributed to the high percentage error. The study is at a conceptual phase so could not attract enough datasets. To further our research, efforts are being made to acquire more datasets from other departments in the university and that of the National Open University of Nigeria. In addition to this, other paraphrases with more datasets would be explored and other semantic similarity measures and techniques such as deep learning would also be harnessed.

References

Abdalgader K. and Skabar, A.(2010). Short Text Similarity Measurement using Word Sense Disambiguation and Synonym Expansion.

Alvi1, F., Stevenson, M. and Clough, P. (2021). Paraphrase type identification for plagiarism detection using contexts and word embeddings. International Journal of Educational Technology in Higher Education 18:42 https://doi.org/10.1186/s41239-021-00277-8.

Antonius, F. Orosoo, M., Saravanan, A., Patra, I. and Prema, S. (2020). Enhanced Plagiarism Detection Through Advanced Natural Language Processing and E-BERT Framework of the Smith-Waterman Algorithm.) International Journal of Advanced Computer Science and Applications14(9), 408-416.

ChmingyangH(2020). An Automatic System for Essay Questions Scoring Based on LSTM and Word embedding. 5th International Conference on Information Science, Computer Technology and Transportation, 355-365. DOI: 1109/ISCTT51595.2020.00068.

Dolan, B., Quirk, C., Brockett, C. (2004) Unsupervised construction of large paraphrase corpora. In Proceedings of the 20th International Conference on Computational Linguistics—COLING ’04, Geneva, Switzerland, 23–27 August 2004; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004; p. 350-es.

Enikuomehin, O. and Dosumu, U (2017), API Assisted Semantic Matching for Subjective Online Examination System. Journal of Research and Review in Science, 4(2017),1–6 DOI:10.36108/jrrslasu/7102/40(0110).

Ferreira, R. Cavalcanti, G., Freitas, F., Lins, R., Simske, S. and Riss, M. (2018) Combining Sentence Similarities Measures to identify Paraphrases. Computer Speech and Language, 47(2018), 59-73. http://doi.org/10.1016/j.csl.2017.07.002

Hazar, M., Toman, Z. and Toman, S. (2019). Automated Scoring for Essay Questions in E-learning. Journal of Physics Conference Series. DOI: 10.1088/1742-6596/1294/4/042014

Lee, M. Chang, J. and Hsieh, T (2014) A Grammar-Based Semantic Similarity Algorithm for

Natural Language Sentences, Hindawi Publishing Corporation Scientific World Journal
http://dx.doi.org/10.1155/2014/437162

Kholodna, N., Vysotska, V., Markiv, O. and Chyrun, S. (2022). Machine Learning Model for Paraphrases Detection Based on Text Content Pair Binary Classification. 4th International Workshop on Modern Machine Learning Technologies and Data Science,
November, 25-26, 2022, Leiden-Lviv, The Netherlands-Ukraine, CEUR Workshop Proceedings (CEUR-WS.org).

Mahmoud, A. and Zrigui, M. (2020). Semantic Similarity Analysis for Corpus Development and Paraphrase Detection in Arabic. The International Arab Journal of Information Technology, 18(1), https://doi.org/10.34028/iajit/18/1/1

Mohamed, M. and Oussalah, M. (2020) A hybrid approach for paraphrase identification based on knowledge-enriched semantic heuristics. Lang Resources & Evaluation (2020) 54:457–485, https://doi.org/10.1007/s10579-019-09446-4

Mwaro, P. N., Ogada, K. andCheruiyot, W. (2020). Applicability of Naïve Bayes Model for Automatic Resume Classification, Int. J. Comput. Appl. Technol. Res., 9(9), 257–264, doi: 10.7753/IJCATR0909.1002.

Nguyen, H. T., Duong, P. H. and Cambrige, E. (2019), Learning Short-text Semantic Similarity with word Embeddings and External knowledge sources. 182(2019), 104842, https://doi.org/10.1016/j.knosys.2019.07.013

Obot, O.U., Udoh, S.S. and Attai, K.F. (2021) 'The suitability of similarity measures to the grading of short answers in examination', Int. J. Quantitative Research in Education, 5(3),.207–222.

Obot, O. U, Onwodi, G. O, Attai, K. F, John, A.E. and Wilson, E (2023). Grading Multiple Choice Questions based on Similarity Measure. Journal of Computer Science and Information Technology, 11(1), 9 -21.

Obot, O, Uzoka, F, John, A. and Udoh, S. (2023). Soft-computing method for settling land disputes cases based on text similarity, Int. J. Business Information Systems, 43(3), 369-393.

Onyshchenko, I., Anisimov, A., Marchenko, O. andIsoieva, M.(2022). Analysis of Semantic Similarity between Sentences Using Transformer-based Deep Learning Methods, Information Technology and Implementation (IT&I-2022), November 30 - December 02, 2022, Kyiv, Ukraine CEUR Workshop Proceedings (CEUR-WS.org), 376-384.

Udoh, S. S., George, U. D., Obot, O. U. and Tom, I. S (2022). Investigation of Similarity Paradigms for Electronic Document Query and Retrieval International Journal of Scientific & Engineering Research 3(4,), 946-959.

Ullah, F., Wang, J., Farhan, M., Habib, M. and Khalid, S. (2018). Software plagiarism detection
in multiprogramming languages using machine learning approach. Concurrency ComputatPractExper. 2018;e5000 https://doi.org/10.1002/cpe.5000.

Ullah – (2021) Software plagiarism detection in multiprogramming languages using machine learning approach. Concurrency and Computation: Practice and Experience - Wiley Online Library'. https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5000 (accessed Sep. 19, 2023).

Vrbanec, T and Meštrovi´c (2020) Corpus-Based Paraphrase Detection Experiments and

Review. Information, 11(241), 1-25; doi:10.3390/info11050241
Vrbanec T. and Meštrović A. (2021), Relevance of Similarity Measures Usage for Paraphrase Detection In Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2021) – Volume 1: 129-138. DOI: 10.5220/0010649800003064.
Wahdan, K. A., Hantoobi, S., Salloum, S. A. and Shaalan, K. (2020). A systematic review of text classification research based ondeep learning models in arabic language, Int. J. Electr. Comput. Eng 10(6): 6629–6643
Yan, S. , Wei, R., Tan, H. and Du, J. (2019). Semantic Document Classification Based on Strategies of Semantic Similarity Computation and Correlation Analysis. Computer Science & Information Technology (CS & IT), NatarajanMeghanathan et al. (Eds) : CSEIT, CMLA, NeTCOM, CIoT, SPM, NCS, WiMoNe, Graph-hoc - 2019 pp. 01-17, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.91301.