LAB-G New Approach for Assessment

Ashish Dwivedi
United Institute Of Technology
Prayagraj, India
Dwivedi_ashish01@gmail.com
Misbah Uddin
United Institute of Technology Prayagraj, India
mohdmisbah5454@gmail.com
Anubhav Kumar Prasad
United Institute of Technology
Prayagraj ,India
anubhavkrprasad@gmail.com
Aman Ojha
United Institute of Technology
Prayagraj, India
amanojha4347@gmail.com
Fahad Bin Zafar
United Institute of Technology
Prayagraj, India
fahadandmuku@gmail.com
Mohd. Alfahad
United Institute of Technology
Prayagraj,India
alfahad2611@gmail.com

Abstract: The most recent Internet innovation has put the entire globe in our hands.Everything proceeds via the Internet, from exchanging information to making purchases. The Internet made the world a small circle. This project is also based on the Internet. This paper shows the importance of chat applications in modern life and their impact on the technological world. The aim of this work is to develop assessment system for students based on Machine learning and Python. The application allows student to take the assessment and get score by machine learning with only a click. This online platform was created to give students the opportunity to communicate or learn under the direction of professors with the aid of assignments. This application was created with a sound architecture for potential future development.

I. Introduction

Lab-G is an Android application used to assist students' growth in lab performance. In this application, faculty monitor the performance of a student and students can see their growth and monitor their lab performance [6]. The app also provides a graphical representation and providesPredictions of students' performance [9]. Here, predication means faculty members can monitor students' learning abilities accordingly. We assist students with their assignment performance. It gives students new experiences and better chances for learning growth. Performing reliable examinations on research material delivered to the labs [2] Lab Growth is an idea regarding creating an app that would basically work as a student progress report [9]. Although evaluation of assessments is a tricky task, mainly because the teacher cannot get the proper criteria through the student area of improvement, Then [10] the data can be compared using tools like concept graphs, feedback sections, latent and quiz scores, and document similarity. The ultimate score can then be determined based on feedback, similarity, and score. Asking for feedback in order to understand people's standards and weak points

is one strategy used to address this problem [11]. A subjective response necessitates the daily assessment of students by the checker. Therefore, handling this important task is much $more effective with resources and time. The evaluation of objective responses by machines [12] is comparatively easy and useful. Evaluation and response can then be aided by an application that receives questions and responses [14]. This research suggests a novel and more learning and natural language processing. two-step approach is used to resolve this issue. The solutions are then evaluated using a range of similarity-based techniques, including word mover's distance, after the responses have first been evaluated using the given keywords. The, result effective approach: automatic evaluation of descriptive question responses using machine learning and natural language processing [18]. A two-step approach is used to resolve this issue. The solutions are then evaluated using a range of similarity-based techniques, including word mover's distance [20], after the responses have first been evaluated using the given keywords [21]. The outcomes of this stage are then used to train a model that can assess responses without the aid of solutions and keywords.

A. Motivation

This kind of automated review significantly improves how the educational sector can carry out its other task and lessens the physical labor required for the mental task, comparing the answers with the correct answer. Teachers and lab instructors use more time in training student, creating good curriculum or openly checking their result.

B. Contribution

The investigation of alternative thresholds for language similarity measurement matrices

C. Solution

- ➔ To show the lab performance of the individual student by checking their growth.
- ➔ To check the Understandability level of the student.
- ➔ It will be helpful for faculty member to assist the student.

D. Current Scenario

We don't have the statistical data for the student lab performance, and don't know about thier actual growth, is they understand or learn in lectures or not.

No one cares about lab or practical learning of the students.

II. Background Review

As it was previously stated, the idea of assessing subjective responses has been researched for around 20 years. This issue has been resolved using the Bayes theorem, K-nearest classifier, massive data natural language processing, similar checker, and even formal methods like formal concept analysis. Statistical, information extraction, and natural language processing are the three main divisions.

III. Literature Survey

They recommended using Semantic Indexing to assess debatable internet queries. They coupled subjective ontologies with string automated[3] segmentation technologies to produce dimensional LSI matrix. The answers were given as 5IDF embedding matrices, and the term-document matrix was then subjected to Singular Value Decomposition (SVD), resulting in a semantic space of vectors. Synonym and polysemy issues were less of a problem because[25] to LSI. Finally, cosine similarity was used to calculate the degree of similarity between the answers.[22] 35 classes and 850 cases from the dataset were rated by teachers.[15] $The findings revealed a 5% discrepancy between teacher grading and the suggested system.(WMD) is a novel tool that Kusner et al. presented for determining the differences between two texts. The system's choice of a lenient WMD technique and the absence of hyper-parameters eased the limitations on the vector space[23]. Real-world[29] datasets were added, such as sentiment analysis from Twitter and BBC[17] sports reports. We investigated numerous techniques previously employed for judging subjective responses and examined their drawbacks.[27] In order to train a machine learning classification model, we used data from

our result prediction module.[19] which we then use to reinforce, is the novel strategy we suggest in this research for solving this challenge. The word2vec word embedding method is used by the result prediction module to build vectors from our data while preserving the semantic meaning of each word. Word Mover's Distance is then used to assess how similar the vectors are to one another. We investigate the effects of applying various machine learning models and contrast our findings with those attained using alternative methods, including Similarity.

III. Block Diagram

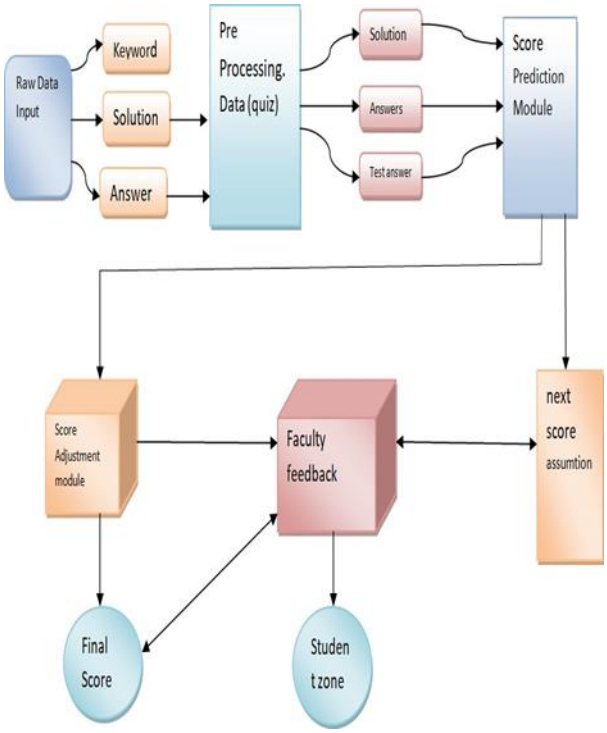

Figure.1

IV. Algorithm

A. Grammar Checker

A key component of natural language processing is the automatic detection and correction of errors seen in English word written by non-native speakers of the language. This uses technology from computer programming. The only way to better language processing and recognition capabilities and enable natural language processing computers to develop into machines with human intelligence is by applying computer programming technology. This is brought on by the vast vocabulary, intricate syntax, ambiguous semantics, and ambiguous speech of the language. There are currently three approaches to fixing grammar mistakes: the rule-based approach, which formulates specific rules to correct specific error types and heavily depends on the quality of the rules while only being able to modify some types of errors; the statistics-based approach, which uses related information like words to obtain text features and model the language and The third method is model-based, which corrects textual problems by utilizing mathematical models. It involves selecting acceptable statistical models to do so. A more accurate and effective method is therefore required to fix the English grammar in this work.

1. Sample Code

```
def GrammerChecker(answer):
        #LanguageTool is open-source grammar
      tool, also known as the spellchecker
```

```
        for OpenOffice. This library allows you
        to  make to detect grammar errors and
        spelling mistakes through a Python script
        or through a command-line interface.
          tool=
language_tool_python.LanguageTool('en-US')
matches=tool.check(answer)
mistakes=[]
for rules in matches:
    if len(rules.replacements)>0:
      mistakes.append(answer[
          rules.offset:rules.errorLength+
          rules.offset
          ])
#Returning No. of  mistakes
return len(mistakes)
 }

 languagetoolpython
  defGrammerChecker(solution) :
  mytool = languagetoolpython.LanguageTool
 ( ' en − US' ) matches = mytool.check(answer)
myMistakes = []
 forrulesinmatches :
       iflen(rules.replacements) > 0 :
       myMistakes.append(answer[rules.offset:
rules.errorLength + rules.offset])
       returnlen(myM istakes)myMistakes = []
```

B. Keyword matching

Some grammar checker software have a feature called "keyword matching for marking" that enables teachers and instructors to grade student's writing based on a list of predetermined words or phrases. This function is frequently employed in educational contexts to evaluate tasks, articles, and essays that need for particular vocabulary or language. The keyword matching feature works by analysing the text and identifying instances where specific keywords or phrases are used. Trainers can compile a list of terms or phrases pertinent to the task or subject, then use the grammar checker to locate these words or phrases in the text. By using keyword matching for marking, educators can assess the students' understanding of the material and evaluate how effectively they have incorporated relevant terminology into their writing. This feature also helps ensure that students have met specific requirements or guidelines for the assignment. #

1. sample code

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
def KeywordMatching(X,Y_list):
result=0
#Tokenizing
X_list=word_tokenize(X)
#List for stopwords
sw=stopwords.words('english')
l1=[]
l2=[]
#remove stopwords from X_list
X_set={w for w in X_list if not w in sw}
#For Y_list
for Y in Y_lst:
   Y_list=word_tokenize(Y)
   Y_set={w for w in Y_list if not w in sw}
#form a set containing both sets:
   rvector=X_set.union(Y_set)
#Preparing two vectors for cosine similarity
  for w in rvector:
       if w in X_set:
           l1.append(1)
       else:
           l1.append(0)
       if w in Y_set:
           l2.append(1)
       else:
           l2.append(0)
  c=0
  #using cosine formula for similarity
  for i in range(len(rvector)):
       c+=l1[i]*l2[i]
    #formula
  #c/sqrt(sum(l1)*sum(l2))
 #Using mathematical equation
    cosine=c/float((sum(l1)*sum(l2))**0.5)
  cosine=cosine*100
  result += cosine
cosine=result/3
if cosine>90:
  return 1
if cosine>80:
 return 2
if cosine>60:
 return 3
```

```
if cosine>40:
  return 4
if cosine >20:
 return 5
return 6
```

C. Similarity checking

Similarity checking for marking is a feature of some grammar checker tools that allows educators and instructors to evaluate the originality of students' writing by comparing it to a database of existing texts. This feature is often used in educational settings to check for plagiarism in written assignments, papers, and essays. The similarity checking feature works by analyzing the text and comparing it to a database of existing texts, including published works, academic journals, and other student papers. The tool then provides a similarity score or percentage, indicating how closely the text matches other existing works. If the similarity score is above a certain threshold, the educator may investigate further to determine if plagiarism has occurred by using similarity checking for marking, educators can ensure that students are producing original work and not simply copying or paraphrasing existing texts. This feature helps promote academic integrity and ensures that students are meeting the requirements and expectations of the assignment.

1. sample code

```
class AnsSim:
def__init__(self,w2v_model,stopwords=None):
   self.w2v=w2v_model
   self.sw=stopwords if stopwords is not None []

def vectorize(self,ans:str) -> np.ndarray:
#Identify vector values for each word in the given Answer
#Param Ans
#return result

   ans=ans.lower()
  words=[w for w in ans.split(“ “) if w not in self.sw]
   word_vcts=[]
   for word  in words:
        try:
            v=self.w2v[word]
            word_vcts.append(v)
       except:
            #Ignore if word does not exist in the vocabulary
   vct = np.mean(words_vcts,axis=0)
   return vct
def _cosine_sim(self,vecA,vecB):
#computing the cosine distance between two vectors
    c=np.dot(vecA,vecB)/
        (np.linalg.norm(vecA)
              *np.linalg.norm(vecB))
    if np.isnan(np.sum(c)):return 0
     return c

def   cal_sim(self,srcAns ,
   tarAns=None,threshold=0):

#compute & revert similarity scores between given source answer & target answer

if not tarAns:
   return []
  if isInstance(tarAns,str):
     tarAns=[tarAns]

  src_vec=self.vectorize(srcAns)
  results=[]
  result=[]
  for s in tarAns:
      tar_vec=self.vectorize(s)
      s_score=self._cosine_sim(
       src_vec,tar_vec)
result.append(s_score)
if s_score > threshold:
    $we can remove this
{results.append(
{“score}:s_score,”ans”:s)
    results.sort(
        key=lambda k:k[“score”]
    ,reverse=True)}
return result
```

V. Module Used

A. Preprocessing Module

Once the user enters their inputs, the solution and the response are both preprocessed. Some

preprocessing methods include tokenization, stemming, stop word removal, case folding, and finding and inserting synonyms word into the normal word(text). Because word2vec has a large dictionary and can make use of such stop words to improve the text's semantic value, it is important to note that stop text are eliminated not when the input is passed to it. Stop words are eliminated before being transmitted to a machine learning model like Multinomial Bayes, despite the fact that they make it more difficult for the computer to recognize patterns.

B. Result predicting module

Centre-piece of study is the Result Predicting Module. Figure 2 depicts how this module operates. It uses the first algorithm listed below: Now that we have the maximum matched solution/answer pair scores, we can calculate the overall score using either WMD or Cosine Similarity. This outcome can be used to calibrate a machine learning model or compared to a real score, is selected.

- To forecast the class or category of the preprocessed answer, a machine learning model uses testing data, and the category is verified using the output of the result prediction module. This increases our level of confidence in the model's predicted outcome.
- The machine learning model receives the preprocessed response and its label. In addition, the model is updated in light of recent data. The final grade, the answer, the response, and the projected class are all sent to the module that predicts the ultimate score. The model is initially fairly sparse and inaccurate, thus it is best to train it with a set quantity of data before beginning to test it.

The last score, the response, the solution, and the anticipated class are given to the last score projection. It is recommended to train the model with a preset set of data before examining it because of the model's initial sparseness and fuzziness. The benefit of the model serving as a confidence builder for

C. Final score predicted module

This figure, which is shown in Figure 2, makes use of the information gained about the class to confirm the final score using the data from the machine learning module. Say the grade is in line with the class. The outcome is regarded as finished. Half of the values in that range are added or deleted if the class does not match the model-suggested score, depending on whether it is higher or lower than the Similarity equivalent score. The adjusted result following the model advised is taken to indicate the final outcome if the machine learning model has undergone substantial training. It is assumed that the score is accurate in Machine Learning Module.

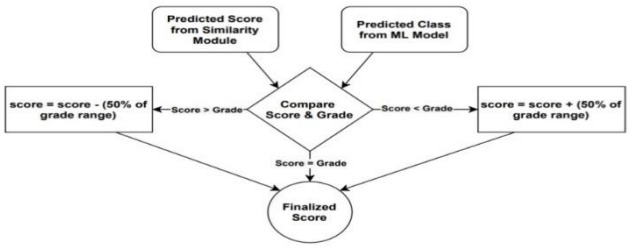

Figure 2.

VI. Precise Result

Judging subjective reactions based on NLP and machine learning approaches. There are two suggested algorithms for predicting scores, and they are up to 80% accurate. To deal with the rare situations of semantically loose answers, many similarity and dissimilarity criterion are examined, along with many other measurements like keyword occurrence and percentage mapped of words. The results of the studies demonstrate that word2vec outperforms conventional word embedding techniques in general because it preserves semantics. Word Mover Distance, which outperforms Cosine Similarity, also speeds up the training of machine learning model. Given

more practice, the model can forecast scores without the need for semantic.

VII. Accuracy Table

Test no.	Actual Score	Predicted Score	Accuracy
1.	50\100	42\100	84%
2.	70\100	62\100	88%
3.	69\100	60\100	86%
4.	88\100	78\100	88%

Overall Accuracy- 86.5%

Description: Test no. is the serial number of tests given by our college and the actual score is the score given by the faculty member which we compare with our module predicted score and we get overall 86.5% accuracy.

1. Outcome

When employed with examine data from the equal dataset used to train the classifiers, assume performs ok. But the predictor consistently misidentifies the contempt-related penalty. This is probably because there aren't many clear examples of contempt in the training and test photos, the data weren't properly labeled before training, and it's hard to recognize contempt in the first place. This is prob- ably because there aren't many clear examples of contempt in the training and test photos, the data wasn't properly labeled before training, and it's hard to recognize contempt in the first place. The classifier is also poor at identifying emotions for test data that comprise expressions that do not obviously belong exclusively to one of the seven fundamental expressions because it has not been trained for other expressions. Future part should concentrate on improving the robustness of the classified by integrating more training photos from different datasets, investigating more prospered detection approaches while preserving computational efficiency, and accounting for the classification of more complex and nuanced expressions. Assessments in school have two functions. With each evaluation, it helps the students demonstrate their understanding, provides feedback for their errors, and offers opportunities for progress.

2. Conclusion

In schools, assessments serve two goals. Each exam provides an opportunity for the students to demonstrate what they have learned, get feedback on their errors, and enhance their performance. Teachers can use it as a fantastic tool to evaluate the efficacy of their preferred method of instruction. What does assessment mean in terms of education? Information is gathered from a variety of sources during an assessment to determine the student's level of understanding and expertise. By going over and examining this data in its context, it could be feasible to establish what worked and what didn't. What function does assessment serve? User can good understand their errors or feedback they got through evaluations. feedback could also shown an anotherchance to adapt the knowledge and again the test to shown better.

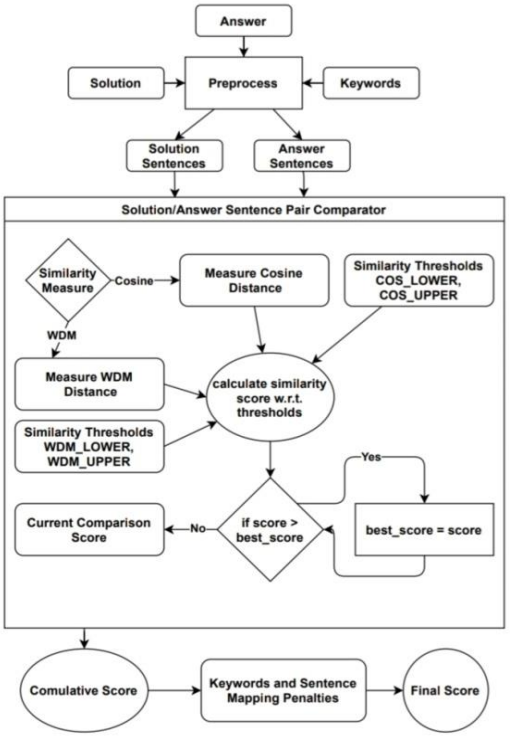

Figure 3.

VIII. References

[1] J. Wang and Y. Dong, "Measurement of text similarity: A survey," Inf., vol. 11, no. 9, p. 421, 2020.

[2] M. Han, X. Zhang, X. Yuan, J. Jiang, W. Yun, and C. Gao, "A survey on the techniques, applications, and performance of short text semantic similarity," Concurr. Comput. Pract. Exp., vol. 33, no. 5, 2021.

[3] M. S. M. Patil and M. S. Patil, "Evaluating student descriptive answers using natural language processing," International Journal of Engineering Research & Technology (IJERT), vol. 3, no. 3, pp. 1716–1718, 2014.

[4] P. Patil, S. Patil, V. Miniyar, and A. Bandal, "Subjective answer evaluation using machine learning," International Journal of Pure and Applied Mathematics, vol. 118, no. 24, pp. 1–13, 2018

[5] J. Muangprathub, S. Kajornkasirat, and A. Wanichsombat, "Document plagiarism detection using a new concept similarity in formal concept analysis," Journal of Applied Mathematics, vol. 2021, 2021.

[6] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in International conference on machine learning, pp. 957–966, PMLR, 2015

[7] C. Xia, T. He, W. Li, Z. Qin, and Z. Zou, "Similarity analysis of law documents based on word2vec," in 2019 IEEE 19th International Conference on Software Quality, Reliability and Security
Companion(QRS-C), pp. 354– 357, IEEE, 2019.

[8] Zhang L, et al. Using linguistic features to estimate suicide probability of Chinese microblog users. In: International conference on human centered computing. Berlin: Springer;

[9] H. Mittal and M. S. Devi, "Subjective evaluation: A comparison of several statistical techniques," Appl. Artif. Intell., vol. 32, no. 1, pp. 85–95, 2018.

[10] L. A. Cutrone and M. Chang, "Automarking: Automatic assessment of open questions," in ICALT 2010, 10th IEEE International Conference on Advanced Learning Technologies, Sousse, Tunisia, 5-7 July 2010, pp. 143–147, IEEE Computer Society, 2010.

[11] G. SRIVASTAVA, P.K.R. MADDIKUNTA, and T. R. GADEKALLU, "A two-stage text feature selection algorithm for improving text classification," 2021.

[12] H. Mangassarian and H. Artail, “A general framework for subjective information extraction from unstructured english text,” Data Knowl. Eng., vol. 62, no. 2, pp. 352–367, 2007.

[13] B. Oral, E. Emekligil, S. Arslan, and G. Eryigit, “Information extraction from text intensive and visually rich banking documents,” Inf. Process. Manag., vol. 57, no. 6, p. 102361, 2020.

[14] H. Khan, M. U. Asghar, M. Z. Asghar, G. Srivastava, P. K. R. Maddikunta, and T. R. Gadekallu, “Fake review classification using supervised machine learning,” in Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part IV, pp. 269–288, Springer International Publishing, 2021.A. Jabbar, S. Iqbal, M. Ilahi, S. Hussain, and A. Akhunzada, “Empirical evaluation and study of text stemming algorithms,” Artif. Intell. Rev., vol. 53, no. 8, pp. 5559–5588, 2020.

[15] N. Madnani and A. Cahill, “Automated scoring: Beyond natural language processing,” in Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018 (E. M. Bender, L. Derczynski, and P. Isabelle, eds.), pp. 1099–1109, Association for Computational Linguistics, 2018.

[16] “TF-IDF,” in Encyclopedia of Machine Learning (C. Sammut and G. I. Webb, eds.), pp. 986–987, Springer, 2010.

[17] G. Grefenstette, “Tokenization,” in Syntactic Wordclass Tagging, pp. 117– 133, Springer, 1999.

[18] K. Sirts and K. Peekman, “Evaluating sentence segmentation and word tokenization systems on estonian web texts,” in Human Language Technologies - The Baltic Perspective - Proceedings of the Ninth International Conference Baltic HLT 2020, Kaunas, Lithuania, September 22-23, 2020 (U. Andrius, V. Jurgita, K. Jolantai, and K. Danguole, eds.), vol. 328 of Frontiers in Artificial Intelligence and Applications, pp. 174– 181, IOS Press, 2020.

[19] A. Schofield, M. Magnusson, and D. M. Mimno, “Pulling out the stops: Rethinking stopword removal for topic models,” in Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers (M. Lapata, P. Blunsom, and A. Koller, eds.), pp. 432–436, Association for Computational Linguistics, 2017.

[20] M. Çagatayli and E. Çelebi, “The effect of stemming and stop-wordremoval on automatic text classification in turkish language,” in Neural Information Processing - 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part I (S. Arik, T. Huang, W. K. Lai, and Q. Liu, eds.), vol. 9489 of Lecture Notes in Computer Science, pp. 168–176, Springer, 2015.(fahad)

[21] J.-E. Kim, K. Park, J.-M. Chae, H.-J. Jang, B.-W. Kim, and S.-Y. Jung, “Automatic scoring system for short descriptive answer written in korean using lexico-semantic pattern,” Soft Computing, vol. 22, no. 13, pp. 4241– 4249, 2018.

[22] M. Oghbaie and M. M. Zanjireh, “Pairwise document similarity measure based on present term set,” Journal of Big Data, vol. 5, no. 1, pp. 1–23, 2018.

[23] K. Orkphol and W. Yang, “Word sense disambiguation using cosine similarity collaborates with word2vec and wordnet,” Future Internet, vol. 11, no. 5, p. 114, 2019.

[24] R. S. Wagh and D. Anand, “Legal document similarity: a multi-criteria decision-making perspective,” PeerJ Computer Science, vol. 6, p. e262, 2020.

[25] M. Alian and A. Awajan, “Factors affecting sentence similarity and paraphrasing identification,” Inter- national Journal of Speech Technology, vol. 23, no. 4, pp. 851–859, 2020.

[26] G. Jain and D. K. Lobiyal, “Conceptual graphs based approach for subjective answers evaluation,” Int.

[27] J. Concept. Struct. Smart Appl., vol. 5, no. 2, pp. 1–21, 2017.

[28] M. Montes, A. Lopez-Lopez, and A. Gelbukh, “Information retrieval with conceptual graph matching,” vol. 1873, pp. 312–321, 01 2000.

[29] V. Bahel and A. Thomas, “Text similarity analysis for evaluation