

# Automatic Exam Generation and Evaluation: A Teachers Toolkit

S A Hariprasad<sup>1</sup>, Darsh Sharma<sup>2</sup>

<sup>1,2</sup>Students, Dept. of Computer Science and Engineering(SCOPE), VIT University, Tamil Nadu, India

\*\*\*

**Abstract:** The project NLP Suite for Teachers is a fully-fledged website that is aimed to help teachers and make their job easy. It consists of basic NLP tools like tokenization, stemming, lemmatization which was the first tool of our website. After that, we built a keyword extraction tool in which teachers can upload answers in the text and the result would be the most important words in that sentence. This can help teachers finding important words without putting in the effort to read the article. We then built a text summarizer in which if text or a corpus is passed the most important sentences will be shown and the tool helps in giving a correct summary of the article's incorrect grammar. In this way, teachers can analyze large texts just by giving them as input to our project tool. Finally, we built a proper quiz-taking platform which is a state-of-the-art technology for teachers these days. With this tool, teachers can create subjective and objective texts with just a few clicks. All they have to do is give a text file with subject articles or modules as an input and automatically questions and answers will be generated randomly. The student can log in with his account and take the test. In subjective tests, keywords from students' attempts and the answers in the database would be checked and hence they would be marked accordingly. The best part is every student will get randomly generated questions relevant to the document provided by the teachers. After te is sent to the teacher with all the students' attempts and results. Hence all the teacher has to do is provide some material for the test and then the platform will automatically generate questions, evaluate answers and store the results as an excel file.

**Key Words:** Natural Language Processing, Question Answer generation, Stemming, Tokenization, Toolkit

## 1. INTRODUCTION

This project aims to help teachers for evaluating answers and conducting quizzes, tests. For this, we have used genism and nltk toolkits. This project includes basic processing steps like lemmatization, stemming, tokenization, etc. It also includes tools that can be used for keyword extraction and text summarization which is useful if a teacher wants to know what are the important concepts in a text corpus. They don't have to read and analyze which part is important. Finally, our project is aimed to create a question answering examination where the teacher just has to input some text data, questions will be automatically generated using the concepts like keyword extraction and text summarization (explained in detail in methodology). Questions can be either subjective

or objective. In subjective questions, the student answers and the answers in the database are evaluated using cosine similarity and according to the distance, results are produced. For the objective part, If the student's answer matches the answer in the database, they will get points. After all, students have attempted a test, their results and scores will be saved in an excel sheet which can be used by the teacher.

### 1.1 Background Study

1)An automatic generator and corrector of multiple-choice tests with random answer

keys:

Study: provides a great idea on how automatically questions is generated and also gives an idea of how to make questions that are relevant to the corpus provided. This paper was able to overcome the previous drawbacks where questions were directly generated by picking lines from the corpus and giving it as questions.

2)Online automatic examination system for digital circuits:

Study: This paper provides an idea of how non theoretical questions can be framed based on the relational database management concept. The given paper also explains how question papers can be generated using PHP and how to have a secure way of conducting exams for the students.

3)Android-based exam paper generator (Android-based E-PAGE):

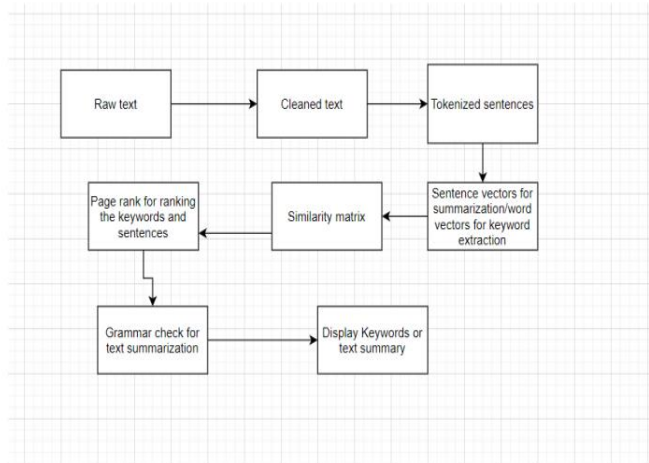
Study: This paper refers to the concepts on how an online examination can be conducted on online devices and also this paper helps in understanding how to develop a website that can be used in both mobile phones and also on computers and how the question can deliver a complete message to the user.

4)Question paper generator and answer verifier

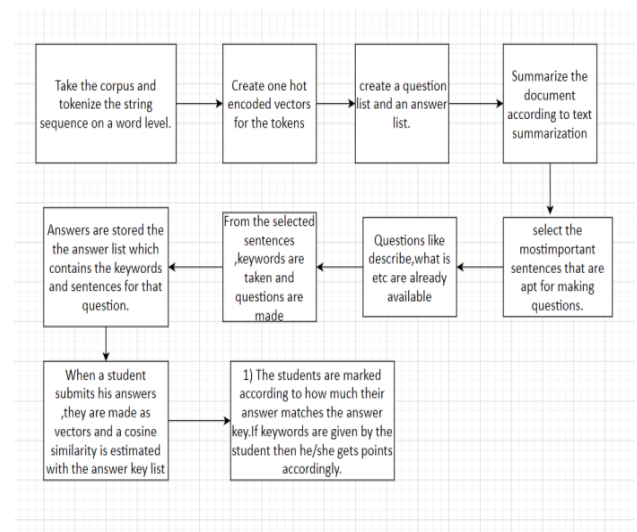
Study: This paper explains how the answer can be verified based on the question and gives an idea of how to make a program that can automatically evaluate the scores and giving the fraction of correctness in the answer using vector processing. This will return the scores with much better accuracy and results compared to directly giving only whole number scores for a given answer.

## 2. ARCHITECTURE DIAGRAM

Keyword extraction and Text Summarization:

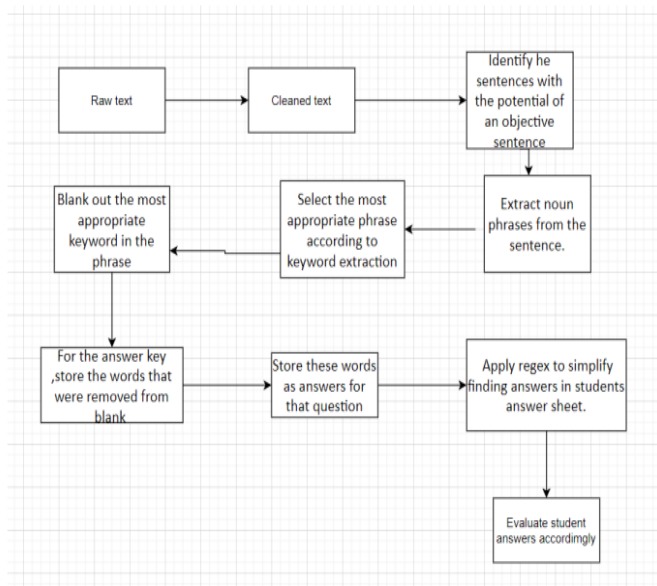


Subjective Tests :



Question Answering Exam

Objective tests:



## 3) Methodology and Proposed model

For the processing tasks like tokenizing stemming, lemmatization, we use basic nltk tools that are really helpful for these tasks. In keyword extraction and text summarization: First raw data is taken and processed, everything is tokenized, stop words are removed, stemming is also done. Sentence vectors are created for text summarization and word vectors are created for keyword extraction. After that similarity matrix is created for these vectors and a page rank algorithm is applied. In that, the topmost words and most important sentences are the output. In-text summarization, important sentences, and keywords are combined with proper grammar to give the output. This is done with the help of the genism library of python. There is no dataset for this, only the text provided by the user is evaluated according to the task Question Answer Examination: Dataset is not used but a text file is provided by the teacher or whoever is creating the exam. The platform will generate questions for the corpus and create an answer key as well. After the student submits the exam, his answers will be evaluated according to the answer key in the list that was created while making questions. Accordingly, the student will get points. Ultimately a pass and fail result is shown with the points that the student has got.

For Objective tests:

- 1) Identify the sentences with the potential of an objective sentence.
- 2) Extract noun phrases from the sentence.
- 3) Apply processing rules, tokenize and grammatize. 4) Select the most appropriate phrase according to the keyword extraction method that was discussed above.

5) Blank out the most appropriate keyword in the phrase, if it can't find any return null.

6) For the answer key, store the words that were removed from the blank. Using wordnet find similar words.

7) Store these words as answers for that question. It's list of questions and answers.

8) Apply regex to simplify finding answers in student's answer sheets.

9) Evaluate the answers accordingly

For subjective tests:

1) Take the corpus and tokenize the string sequence on a word level.

2) Create one-hot encoded vectors for the tokens and create a question list and an answer list.

3) Summarize the document according to the text summarization technique discussed above and select the most important sentences that are apt for making questions.

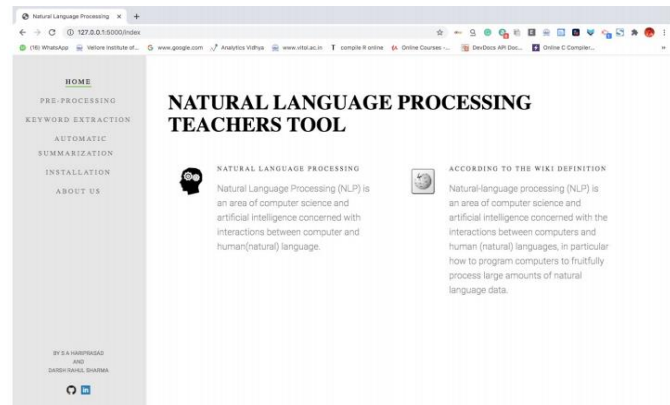
4) Questions like describe, what is etc. are already available.

5) From the selected sentences, keywords are taken and questions are made. eg in a document for NLP, a question can be what is NLP

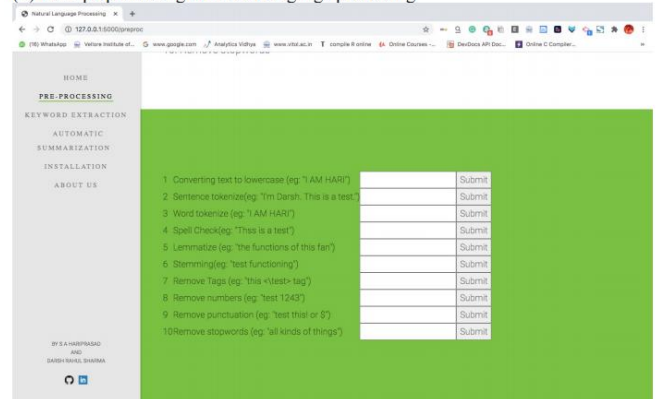
6) Answers are stored in the answer list which contains the keywords and sentences for that question.

7) When a student submits his answers, they are made as vectors and a cosine similarity is estimated with the answer key list. The students are marked according to how much their answer matches the answer key. If keywords are given by the student then he/she gets points accordingly.

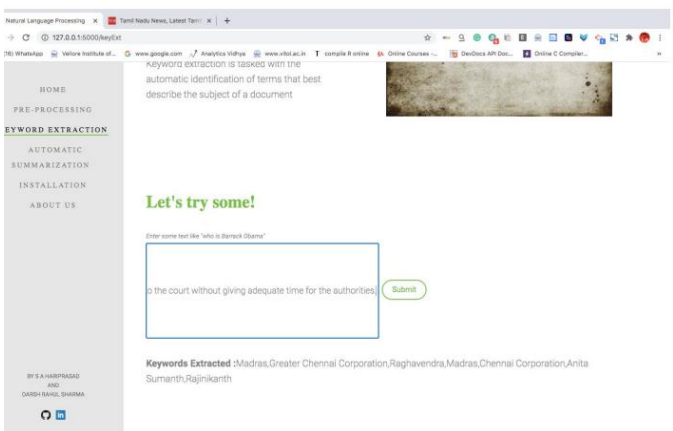
#### 4. Outputs



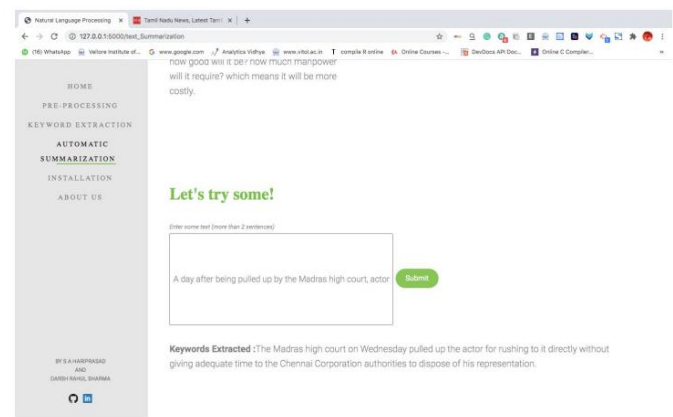
(ii) basic preprocessing in natural language processing



(ii) Keyword extraction

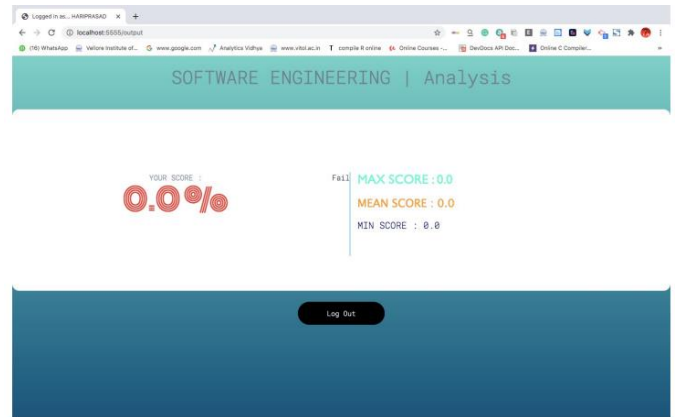
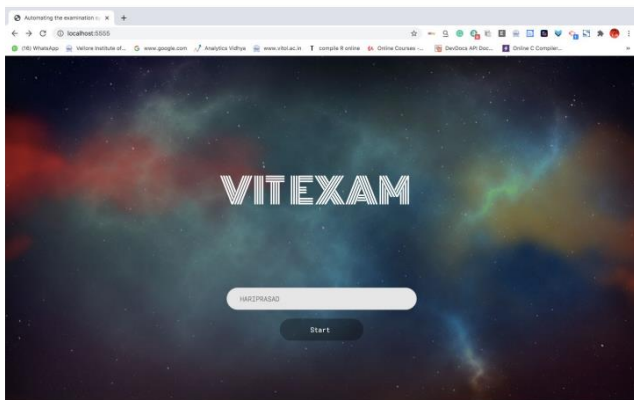


(iii) Automatic summarization

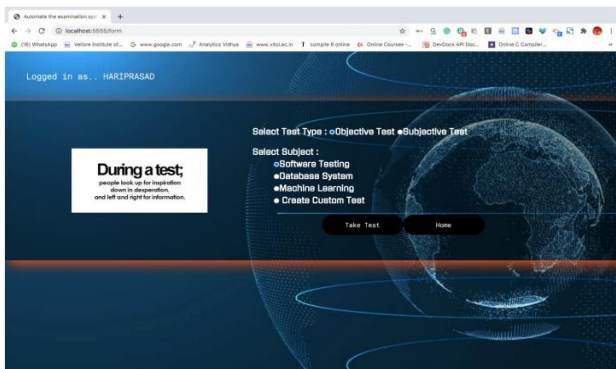


Part2-question and answer generator and evaluator

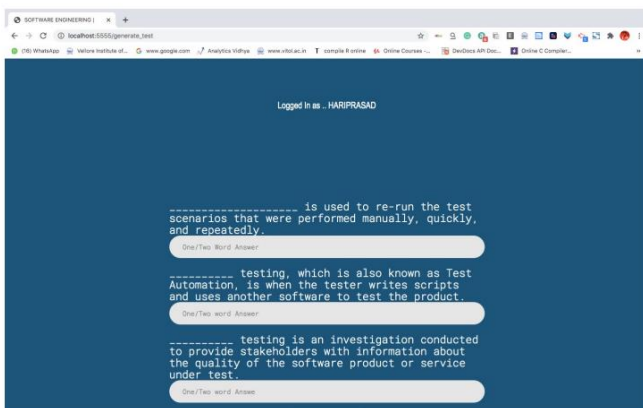
(i) Home page



(ii) Choose which test to attend



(iii) Test page to see and attend the quiz



5. CONCLUSION

So in this project, we made an application for teachers. Tools like keyword extraction, text summarization for understanding large text corpus can be used. In the question answering system that we built, students can attend objective and subjective tests for the corpus uploaded by teachers. The questions will be generated on their own and the answers from the students will be evaluated by the system itself. So it is an end-to-end application that is aimed to reduce teacher's workload. We have built this project from scratch using NLP algorithms and tools. The interface is made to make the users easily access our tools and use the system to its fullest capabilities.

REFERENCES

- [1] Rjoub, B. Tall, N. Sharou and L. Mardeeni, "A Novel Multi-Forms Multiple Choice
- [2] Editor Exam Tool Based on HTML Website," 2006 7th International Conference on
- [3] Information Technology Based Higher Education and Training, Ultimo, NSW, 2006,
- [4] pp. 854-869, doi: 10.1109/ITHET.2006.339710.
- [5] C. Vafeiadou, P. Vasiloudis and M. Dasygenis, "Online automatic examination
- [6] system for digital circuits," 2016 5th International Conference on Modern Circuits
- [7] and Systems Technologies (MOCAST), Thessaloniki, 2016, pp. 1-4, doi:
- [8] 10.1109/MOCAST.2016.7495142.F. de Assis Zampiroli, V. Batista and J. A. Quilici-Gonzalez, "An automatic

- [9] generator and corrector of multiple choice tests with random answer keys," 2016
- [10] IEEE Frontiers in Education Conference (FIE), Erie, PA, USA, 2016, pp. 1-8, doi:
- [11] 10.1109/FIE.2016.7757467.
- [12] M. Fatangare, R. Pangare, S. Dorle, U. Biradar and K. Kale, "Android based exam
- [13] paper generator (Android based E-PAGE)," 2018 2nd International Conference on
- [14] Inventive Systems and Control (ICISC), Coimbatore, 2018, pp. 881-884, doi:
- [15] 10.1109/ICISC.2018.8398926.
- [16] P. Pisat, D. Modi, S. Rewagad, G. Sawant and D. Chaturvedi, "Question paper
- [17] generator and answer verifier," 2017 International Conference on Energy,
- [18] Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp.
- [19] 1074-1077, doi: 10.1109/ICECDS.2017.8389603.