**QUESTION GENERATION USING NLP**

This paper proposes a method to generate questions using coreNLP and tree regular expressions for parse tree manipulation

Nowadays, professors spend a lot of time generating test papers and quizzes manually. Similarly, students spend a lot of time on self-analysis. Moreover, students are dependent on their mentors for the self-analysis. Hence, working on this NLP area, has a huge scope of development at this moment. We want to build a computer application system that can help you in calibrating yourself and remove any dependencies on mentors. Here, students can give the input text of whatever material they referred to, and on this basis they get a set of questions with answers from which they can do a self-analysis. A similar approach is used by mentors for creating test papers and quizzes.

Moreover, online examinations have become very popular, including many major examinations, such as GATE, CAT, and NET. Multiple Choice Questions (MCQ) is very easy for evaluations, and its evaluation is implemented through computerized applications so that results can be declared within a few hours, and the evaluation process is 100% pure.

By making this computerized application, we can reduce the task of an educator. Much time can be saved if we can know what appropriate questions can be asked for the given input of text.

Hence, we want to develop a system which can generate various logical questions from the given text input

PROPOSED METHODOLOGY

First, the input text that is the paragraph is split into single sentences, then using appropriate question word, we generate questions. From a single sentence, we produce yes-no questions over the sentence and WH questions from the subject, object, adverbials, and prepositional phrases in the sentence.

IMPLEMENTATION

* Sentence pre-processing: Questions generated from large complex sentences can sometimes be meaningless which is why we need to preprocess them. In the pre-processing stage, complex sentences are divided into simpler sentences.
* Parsing: Firstly, we parse each input sentence using the Stanford parser.
* Extracting Independent Clauses from Complex Sentences: We split one type of complex sentence: those formed by joining simple independent clauses using conjunctions.
* Resolving Anaphors and Replacing Pronouns: -A complex sentence includes pronouns and simply splitting the sentences gives ambiguous or unclear output. Anaphora resolution and pronoun replacement enhance the quality of questions generated from such sentences.
* Additional Sentence pre-processing: In some of the question generation modules, we use the following processes question word identification; subject-auxiliary inversion; appositive removal and prepositional phrase removal.
* Question Word Identification: The Stanford Named Entity Recognizer (NER) is used in this to find the entity type of each answer phrase we find. The Stanford named entity recognizer outputs three entity types as well as NO ENTITY: PERSON, LOCATION, and ORGANIZATION. In addition to this, we used spaCy which is able to identify many more entities which help us identify question word more accurately. We select the question word using the identified entity type
* Subject-Auxiliary Inversion (SAI): Subject-auxiliary inversion is used to construct Yes-No questions
* Question Generation: After each sentence is pre-processed, question generation is performed on each independent clause.

CONCLUSION

This system can be used in multiple self-analysis scenarios. For example, students can use it to make learning easier as well as more interactive and interesting. Teachers and professors can use this system to quickly create a quiz. A central examination board can use this system to generate a unique test that is not known to any professor, eliminating the possibility of cheating and thereby securing the privacy and integrity of the examination. Question generation system always produces more questions than the number of questions generated by the human judges.