VIRTUAL ASSISTANT FOR TEACHERS

A BE-PROJECT REPORT

By

Princeton Baretto (8316)

Carol Sebastian (8320)

Sherwin Pillai (8358)

Under the guidance of

Prof. Supriya Kamoji

DEPARTMENT OF COMPUTER ENGINEERING

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

FR. AGNEL ASHRAM, BANDRA (W),

MUMBAI - 400 050.

UNIVERSITY OF MUMBAI

(2020 – 2021)

FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERING

FR. AGNEL ASHRAM, BANDRA (W),

MUMBAI 400 050

CERTIFICATE

This is to certify that the following students working on the project “Virtual Assistant” have satisfactorily completed the requirements of the project in fulfillment of the course B.E in Computer Engineering of the University of Mumbai during academic year 2020-2021 under the guidance of “Prof. Supriya Kamoji”.

Submitted By: Princeton Baretto (8316)

Sherwin Pillai (8358)

Carol Sebastian (8320)

Prof. Supriya Kamoji

Guide

Dr. B.S.Daga

Head of the Department

Dr. Srija Unnikrishnan

Principal

CERTIFICATE

This is to certify that the project synopsis entitled “Virtual Assistant For Teachers” submitted by the following students is found to be satisfactory and the report has been approved as it satisfies the academic requirements in respect of BE Project work prescribed for the course.

Princeton Baretto (8316)

Sherwin Pillai (8358)

Carol Sebastian (8320)

Internal Examiner:	External Examiner:
(Signature)	(Signature)
Name:	Name:
Date:	Date:

Seal of the Institute

DECLARATION OF THE STUDENT

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature of the student Date:

(Princeton Baretto) (8316)

Signature of the student Date:

(Sherwin Pillai) (8358)

Signature of the student Date:

(Carol Sebastian) (8320)

ABSTRACT

The project tries to develop an online platform that facilitates the existing traditional processes by introducing a virtual assistant to give support to our information technology program. We believe that the whole Question-Paper generation and assignments grading process could be a tedious job for the teachers.

Our system puts forward a solution to automatically generate questions and provide sample answers for the same from a given document. Based on these generated Q&A, it could also grade a student's handwritten answer sheet/assignment. Finding internal information from a PDF or docs should be easier. We believe that everyone should be able to use modern search technologies to find information in their documents. Our Virtual assistant allows the teachers to ask a question in natural language and get an answer without having to read the internal documents relevant to the question, making it easy for them to prepare notes for their upcoming lectures.

Table of Contents

List of Figures

Chapter 1

Introduction

Question generation (QG) problem and Question Answering (QA), which takes a context text and a supporting phase as input and generates a question or answer corresponding to the given supporting phase, has received tremendous interest in recent years from both industrial and academic natural language processing communities. There's always a mess when teachers have to prepare Questions or any of the students or the teachers or literally any person has to understand a certain pdf or content of the text. Right from various research papers to any textbook chapter pdfs or text files, to understand or query the important questions you have in mind manually is a difficult task. Apart from this, the process of making the questions from those pdfs or text content is another difficult task since the teacher has to go through the entire file and then make or generate questions. Also, there is a lot of human thinking involved and also a lot of time is consumed in the whole process.

1.1 Aim:

Our solution would help them understand the pdf or the text by asking questions on their text and our model. Also apart from this, the user can also automatically generate relevant questions from the file that is uploaded by them.

1.2 Objective:

The system aims to develop a responsive system for Question Generation and Question Answering. The system can be used in educational institutions which would help the teachers as well as students to understand and study files easily and efficiently.

1.3 Scope:

We aim to focus on the following aspects through the course of our project:

- Answer Extraction from the file.
- Question Generation
- Closed Domain Question Answering

Chapter 2

Review of Literature

QA1: EDUQA: EDUCATIONAL DOMAIN QUESTION ANSWERING SYSTEM USING CONCEPTUAL NETWORK MAPPING

Introduction: This paper reviews the concept of dividing the whole process into 3 chunks of processes
1:Entity recognition
2:Question Analysis that filters relevant features
3:Answer Retrieval for extracting the answer based on the above two processes.

Summary: Entity Recognition is done using the DCN (Dynamic Concept Network). the main task of this module is to extract entities and their relationships
The list of the entities is passed to the Question Analysis module.
This module takes the input as a question and then tokenizes it and extracts the longest prefix sequence from the entity list provided by the DCN module.
After this, the information is passed on to the answer retrieval module which tries to match the entities with the most relevant relationship i.e it extracts the
relationships from the entities found by the Question analysis module. If we find any relation then it is passed to the concept network to extract the answer and
then it's given back to the user.
But if we don't find a relation, then the person has to find the answer and mark it and this is updated in the concept network. This process is called fly learning.

Conclusion: The model was able to correctly answer 80% of the definition based questions and 65% of the other type of questions.

QA2: A Novel Approach for Semantic Similarity Measurement for High-Quality Answer Selection in Question Answering using Deep Learning Methods

Introduction: This paper proposes a complete neural-based architecture for the QA models i.e there is no NLP included and all the understanding is done
solely by the neural network.
In this paper, there are three steps too:
1: Question analysis
2: Document retrieval
3: Ranking of Answers.

Summary: Here, first the questions are converted into word vectors using the word embedding matrix, they have used and tried different word embedding matrices
like Word2Vec, fastText, Glove, Baroni, SL999. Among these, the highest accuracy was gained by the SL999 matrix due to its wide range of word reach.

The next step of this pipelines consists of 2 important parts of the pipeline, the prediction is done using two techniques mainly Versatile global T=max pooling and
Deep LSTM. Once the prediction is obtained based on the model it is given to the efficient DFM which filters the answers based on ranks. The model was trained on
4 different types of datasets like STSB, Wikipedia QA dataset, MRPC, and SICK datasets. the average accuracy scored on all of them was about 82-83%

And also the maximum accuracy is always obtained when we use the SL999 word embeddings.

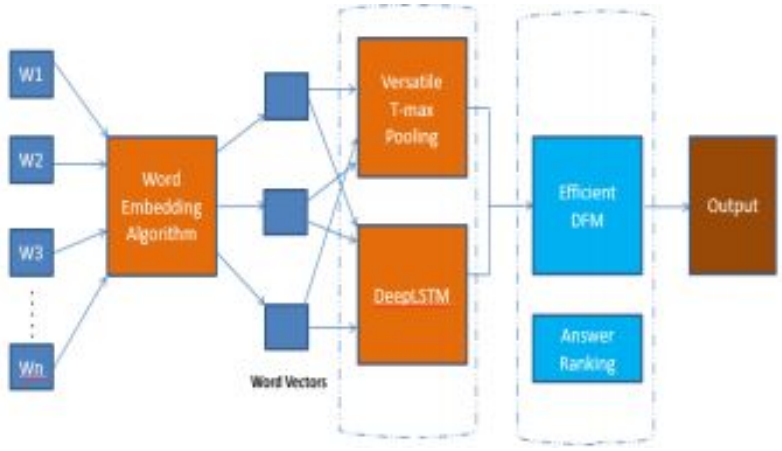

Fig 1.0 Answer Selection Proposed Model

Conclusion: In this Research paper, they provided a Versatile global T-max pooling and DeepLSTM for quality answer prediction.
They have additionally used Efficient DFM to forecast the nice solutions and specially DFM is used for the ranking cause.

QA3: Semantics-Enhanced Answer Selection in Closed-domain Question Answering System

Introduction: This paper stresses the Information retrieval task in question answering systems**.** The overall system architecture of the proposed system consists of three main components namely Domain Resources, Corpus, and MPS Components.

Summary: The Domain Resources refer to the Knowledge and the Database one currently possesses related to the Domain. These act as a backend to the QA system. The Domain Dataset is the document that is to be queried by the users. It can be in any text format.
The next step in this pipeline is the Corpus, which provided knowledge about the named-entity relation to the model. Here the features of the query are extracted and then given to the next phase of the pipeline that is the MPS(Most probable Sentence). As we know that the features may

correspond to various sentences, here the MPS selected the most probable sentence as its name suggests. The features of the query are compared against those of all the sentences in the Domain Dataset and return the best sentence to the user. The first model is the Naive Bayes classifier just classified the question into some types of questions so that it is easy for the MPS to find the sentence.

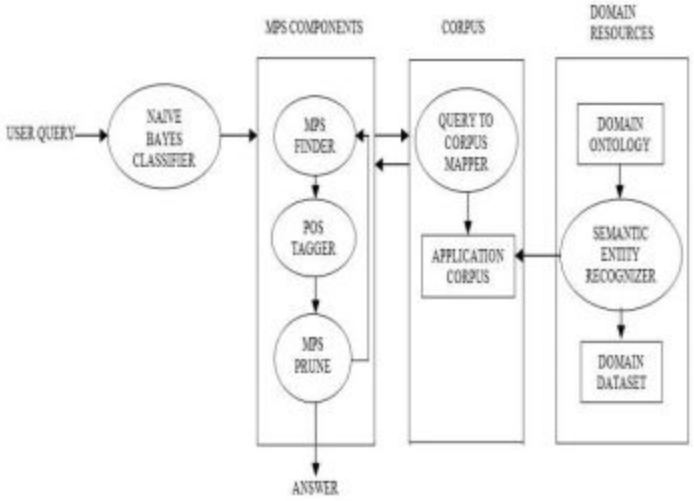

Fig 2.0 Semantics-Enhanced Answer Selection Architecture

Conclusion: The complete process obtains an overall accuracy of 71% on the real-life data as tested on 500 Wikipedia questions that were manually prepared. This system offers the betterment of user requests along with the improved specification and matching techniques.

QA4: Semantics-Enhanced Answer Selection in Closed-domain Question Answering System

Introduction: This paper proposes an architecture for retrieval of answers based on many documents, for example, you have 10 documents and you ask a question, the model will search the most relevant document as well as the most relevant answer from those documents.

Summary: The first step in this pipeline is to receive the question from the user, after this the type of answer it wants is detected. Once we have this it

extracts the names and the entity from the query and detects the keywords from the query. Once we have all this information some NLP techniques are used to get the meta-information too and the final query is formed bringing all this together. This metadata is matched with the documents we have and the most relevant documents are extracted using this metadata. After this, the most relevant passages are retrieved from those documents and every passage is given a score based on their relevance to the query. This is passed to the Local Proximity Prioritizer. The goal of this component is to highlight candidate answers that show a denser distribution of matches in their text.
After this, to filter out more answers we pass it to the Keywords Overlapping module. Here, a metric is used to measure the similarity between texts is defined, which assesses the percentage of overlapping words between the candidate answer and the question text. If the overlapping percentage is greater than a configurable threshold, the score for the candidate answer is increased. The default threshold value is 33%.

This is one of the final stages of the pipeline. Once we have obtained the filtered answer, we pass it to the name entity module which matches the NE of the query and the answer. In case of a match, the score is incremented, giving more relevance to the candidate answer. Finally, the answer with the highest score is obtained.

Conclusion: The framework is able to retrieve a correct answer for 80% of the questions in the dataset. Precisely it gives the right answer, as the first answer, in 66% of the cases.

QA5: Learning to Rank Answers to Closed-Domain Questions by using Fuzzy Logic

Introduction:

Question answering (QA) is a challenging task and has received considerable attention in the last years. Selecting one or more answers from a list of candidate answers has been given utmost importance. This paper proposes a fuzzy approach for ranking and selecting the correct answer among a list of candidates in a state-of-the-art QA system operating with factoid and description questions on Italian corpora pertaining to a closed domain.

Paper Summary:

In detail, the proposed approach is able to learn fuzzy rule-based models, by means of which the scores, measuring answer evidence affected by uncertainty and vagueness, are transformed into a unique confidence grade, according to which the answers are given a ranking order, which is used to choose the correct answer. Fuzzy ranking models can explicitly show which measures better explain differences between correct and incorrect answers, depending on the question type. The Architecture employed here consists of 4 modules:

1. **Indexing**: It provides the documents with annotations and respective indexing.
2. **Questions Processing:** This phase processes the question text and extracts a set of features, to generate a query for the IR engine. It has components like ANswer Type Detection, Stop Word Removal, NLP Metadata Extraction, and Query Formulation.
3. **Information Retrieval:** The search engine retrieves a set of documents satisfying the query, and spits them into sentences.
4. **Answer Selection:** This module is made of a collection of components, each one assigning a score to the candidate answers. These components include Named entity matching, Keywords Overlapping, Proximity with Gap, Proximity Distance Highlighted Text to LAT, Multiple LAT discourager.

Results:

Five fuzzy ranking models are learned from the supervised questions set, one for each question type; therefore, depending on the question type, a fuzzy model based on single scores is chosen for ranking candidate answers to each question.

TABLE II. ACCURACY@1 OF ANSWERING CORRECTNESS

Method	Question Type					All questions
	Description	*Date*	*Entity*	*Location*	*Person*	
Empirical averaging	0.85	0.75	0.80	0.80	0.85	0.79
LFA	0.91	0.84	0.81	0.84	0.84	0.83

From the above table, it can be evinced that using the proposed approach for ranking answers instead of empirically combining single scores results for most of the question types in an accuracy improvement. If all 907 questions of the dataset are considered, the usage of the proposed approach enables us to answer correctly to 756 instead of 721 questions. In detail, the Likelihood-Fuzzy Analysis has been applied in order to learn fuzzy rule-based models able to discern correct (True) from incorrect answers (False). The different scores, affected by uncertainty, have been transformed into a unique confidence grade, according to which the answers have been ranked in order to determine the best candidate as the first-ranked one.

QA6: Question Answering System on Education Acts Using NLP Techniques

Introduction:
This paper proposes an architecture for the Question Answering System on Education Acts using NLP. This paper uses CDQA as it gives better accuracy than Open Domain Question Answering Systems.

Architecture:
An input for the proposed system will be a query related to education acts or different information related to education. For example “What is the duty of parents to secure the children's education?”, “What are the funding

authorities of school?” The Question keyword is calculated by removing stop words and performing stemming on questions to extract the answer. Metadata will be generated for the dataset related to Education Acts. Using these keywords, the original passage or sentences are tagged to give candidate answers from the answer extractor. According to the given question, the highest score candidate answer will be shown as the final answer. The system will produce the accurate answer for trained questions and then will test to measure the accuracy of untrained questions.

Conclusion:

The QA system for closed domain of documents related to education acts using NLP techniques and information retrieval are proposed to give the accurate and suitably more correct answers for user’ queries.

The following is the architecture diagram for the proposed system:

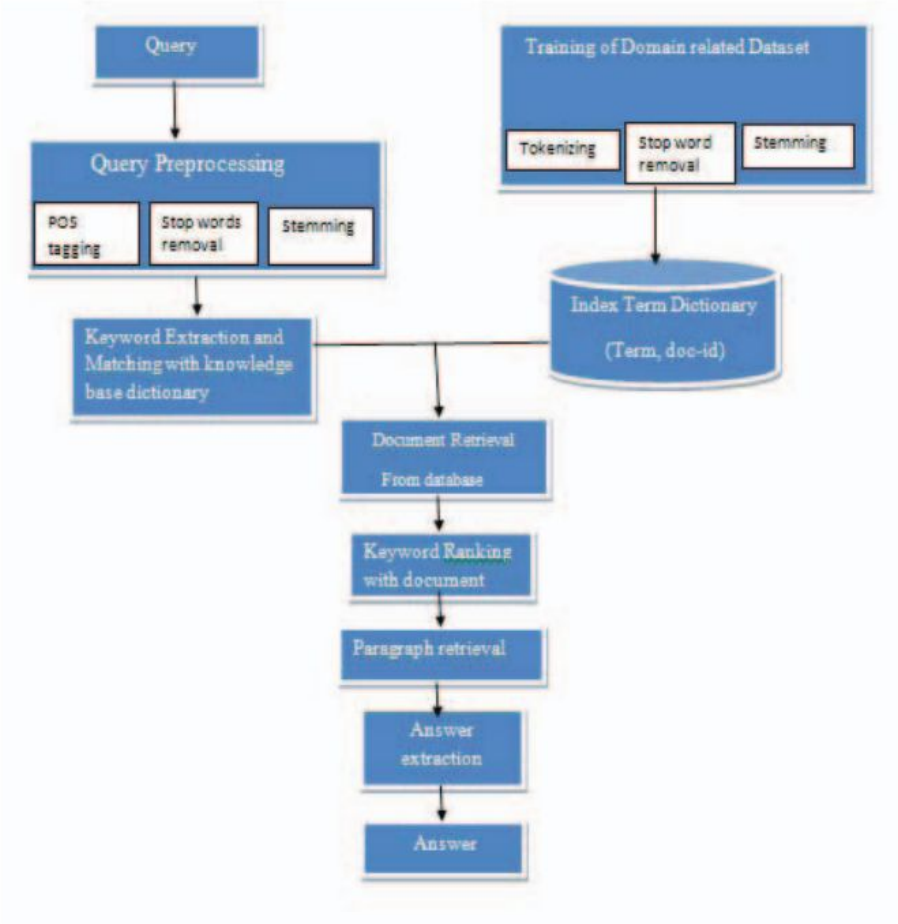

Fig 3.0 Question Answering System on Education Acts

QA7: Evaluating Reasoning in Factoid based Question Answering System by Using Machine Learning Approach

Introduction:

The main purpose of this paper is to provide a Factoid based CDQA system using Machine Learning. Mainly Factoid based Questions consist of 6 types , Wh type like where, which, when, what, why etc. definition questions, Yes-No/True-False Questions, instruction-based questions, explanation questions and list questions. Their system proposes an architecture to solve all types of Factoid Questions.

Summary:
The Proposed Model of the system is :

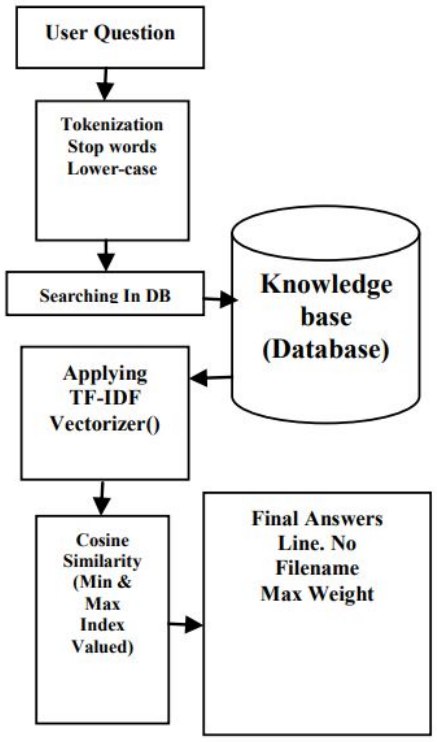

Fig 4.0 Evaluating Reasoning in Factoid based Question Answering System

When a user gives a query, the system pre-processes the given question into tokenization, stop-words, converting the given question into a lower-case. After cleaning, the system searches the words in the knowledge-based file (database) for example why Shivaji was a great man? After preprocessing the words remaining is Shivaji great man. It is searched in the database. If the word is found in the database then it is kept in the list and further TF-IDF vectorizer improved algorithms are applied to it. The maximum scored candidate answer is declared as the final answer.

Conclusion:
The following is the result of the proposed system:

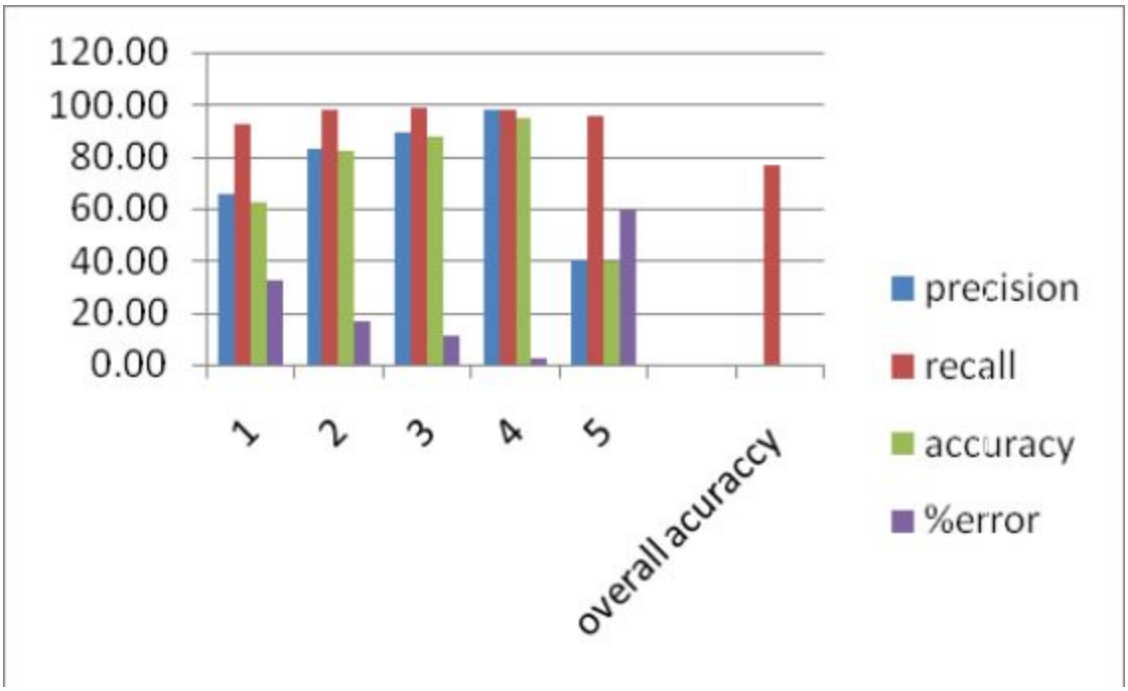

Fig 5.0 Metric Evaluation of Proposed System

They tried to develop a Question Answering System using TF-IDF vectorizer using the Scikit-learn library of Machine Learning. Overall accuracy of their system is 76.32 %.

QA8: Questionator - Automated Question Generation using Deep Learning

Introduction:

The paper proposes a state-of-the-art solution using a pipeline that utilizes natural language processing and image captioning techniques capable of generating questions not only for textual but also for visual inputs. Along with the question, distractors for the generated questions and their answers are also created.

Paper Summary:

Section I	Image Captioning
Section II	Question Generation.
Section III	pipeline of proposed system.

Image Captioning

The image captioning module converts a given input image into a natural language description. The natural language provides a good solution for describing the semantic information of images. The caption generated by the image captioning is now fed through the question generation.

Question Generation

The question generation is performed using sentence splitting and Named Entity Recognition. The caption obtained from the previous module is passed through a dependency parser (Stanford CoreNLP), that extracts the subject, object and the predicate in the sentence. These 3 subparts of a sentence are then passed through a POS (Part Of Speech) tagger and a “Wh” question is generated.

MCQ Generation

It suggests question generation as a means for question answering. This module makes use of GloVe to make a Vector representation for words and choose the distractors for the generated questions.

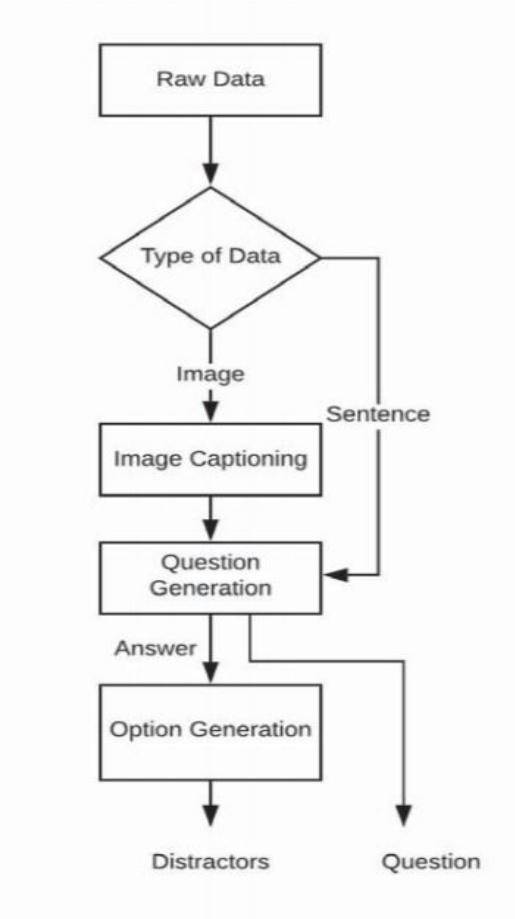

Fig 6.0 Flowchart for Questionator - Automated Question Generation

Conclusion

The proposed question generation pipeline has demonstrated that automated generation of questions can be found in the education domain for generating Multiple Choice Questions(MCQ). This is a robust system that not only works for sentences but also for images. Further additions to the captioning dataset , for instance adding complex sentences and then improving the question generation model to generate questions on these complex sentences could yield better results.

QA9. Deep Neural Network Models for Question Classification in Community Question- Answering Forums

Introduction:

In the case of an opinion-based or a yes/no question that wasn't previously answered, an external knowledge source is needed to generate the answer. The paper proposes a LSTM based model that performs question classification into the two aforementioned categories. Given a question as an input, the objective is to classify it into opinion-based or yes/no question. The proposed model was tested on the Amazon community question-answer dataset.

Paper Summary:

Section I	Learning, Soft Computing
Section II	Approach-1 (CNN)
Section III	Approach-2 (LSTM)

Learning, Soft Computing

Identifying such duplicate and near-duplicate questions automatically and later, automating answer generation for them. When the question is opinion-based or is not currently addressed in the system will the involvement of a human or customer care professional be required.

Approach-1 (CNN)

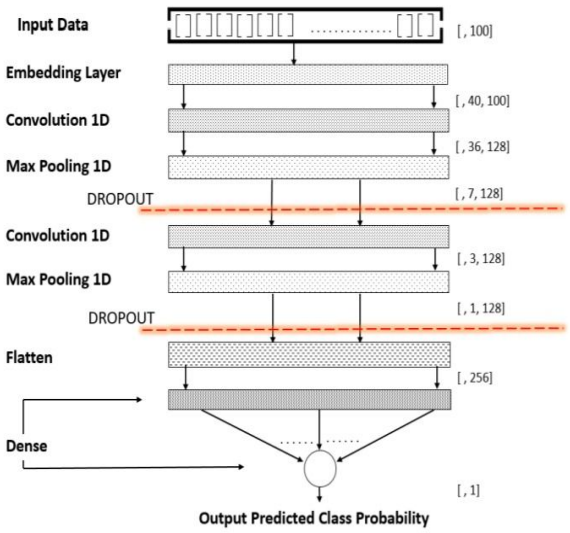

Fig 7.0 Deep Neural Network Models for Question Classification (CNN)

Approach-2 (LSTM)

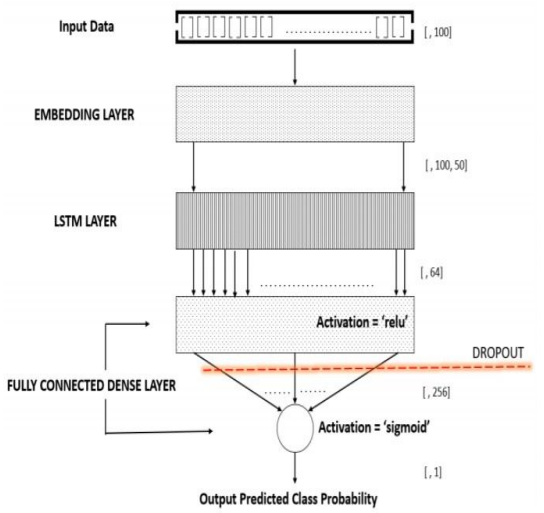

Fig 8.0 Deep Neural Network Models for Question Classification (LSTM)

Conclusion

The CNN model achieved a maximum accuracy of 72.37% for the electronics category of the dataset, but suffered from overfitting problems. The LSTM model was able to fit the data really well and outperformed CNN by a significant margin by achieving an accuracy of 93.4% for the electronics category of the dataset.

QA10: QuGAN: Quasi Generative Adversarial Network for Tibetan Question Answering Corpus Generation

Introduction:

The paper aims to generate a QA corpora for low-resource languages, such as Tibetan. The proposed system is a QA corpus generation model, called QuGAN. This model combines Quasi-Recurrent Neural Networks and Reinforcement Learning.The QRNN used as a generator for GANs, which speeds up the generation of text. At the same time, the reward strategy and Monte Carlo search strategy are optimized to effectively update the

generator network. Finally, we use the Bidirectional Encoder Representations from Transformers model to correct the generated questions at the grammatical level.

Paper Summary:

Section I	Generator
Section II	Discriminator
Section III	Reinforcement Learning Optimization
Section IV	Grammar Optimization
Section V	Answer Matching

Generator

Before the model training, MLE(maximum likelihood estimation) on randomly sample data to generate questions more efficiently i.e To derive the maximum probability text sequence. We use the QRNN model as a generator. In QRNN, there are three pooling modes : f-pooling, fo-pooling and ifo-pooling. This paper uses f-pooling.

Discriminator

This paper uses the fundamental LSTM model as the discriminator to judge if the generated text is real. The discriminator scores the generated sequence in whole sentences and feeds the scores back to the generator.

Reinforcement Learning Optimization

Traditional Monte Carlo search is very time consuming. It needs to generate samples every time. Meanwhile, it calculates the previously generated item when calculating some of the later partial sequence reward estimates, resulting in overfit. Therefore, they optimized the Monte Carlo search algorithm and scored the next sequence through the generated partial sequences, so the score of the entire sequence can be quickly obtained.

Grammar Optimization

To eliminate the grammatical errors of the sentences generated by QuGAN, this paper uses the BERT model to modify and optimize the questions, including two parts: random mask and next sentence prediction.

Answer Matching

They used semantic similarity to match question and answer in corpus. The answer with the highest similarity of the question is the answer to the question.

Architecture

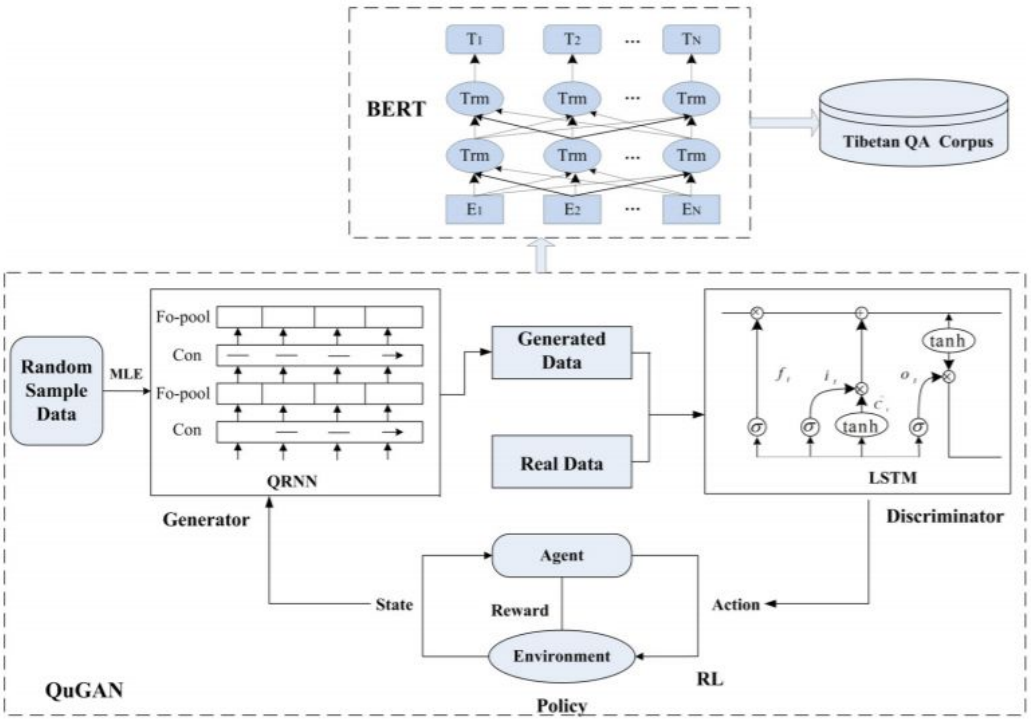

FIGURE 1. The framework of QuGAN model.

Fig 9.0 QuGAN Architecture

Results

Random initialization	MLE	Generated by Our Model
སྨན་ངག་གནམ་གཤིས་ཚ་ཚང་།	ཅི་ཞེག་ལ་སྨན་ངག་ཅེས། $ སྨན་ཞིང་ཚིག་འཛེའམ་པ་ཞིག་ལ་ཟེར།	ཅི་ཞེག་ལ་སྨན་ངག་ཅེས། $ རྩོམ་པ་རིགས་ཤིག
(Poetry, weather, complete.)	(What is poetry? $ Beautiful words, words, sentences.)	(What is poetry? $ A kind of writing.)
དཔྱོད་ཚིན་ཞིས་པོས་དེ།	དཔེ་འབྱོར་དང་དཔྱོད་ཚེས། $ ཐོན་འཇུག་དང་ཚེས་འཇུག	ཅི་ཞེག་ལ་བཅྱོད་ཚེས། $ གནས་དང་ཁེད་བའི་རྣམ་འགྱུར་མི་གསལ་ཞིང་། དེད་རྟེན་ཡང་ཡང་བཞིན་ནས་མིང་དང་ཚིག་གི་འཇོད་པའི་སྐྱབས་བོར་འཕྲིན་རྒྱབ་ཞིག་ལ་ཟེར།
		(What is a consonant letter? $ It does not make sense in itself, encountering the base word can be fully issued corresponding tone.)
ཅི་ཞེག་ལ་རྩ་བ་ཟེར།	ཅི་ཞེག་ལ་རྩ་བ་བཞི་ཟེར། $ མ་བྱིན་ལེན། བསྲུན། རྫོག་གཙོང་དེ་བཞི།	ཅི་ཞེག་ལ་རྩ་བ་ཟེར། $འཇ་དངོས་ཀྱི་འཛུང་རྩ་དང་གཞི་རྩ་ལ་ཟེར།
(What's fundamental?)	(What are the four fundamentals? $ No lying, no harm.)	(What's fundamental? $ The root or basic source or basic of the physical object.)

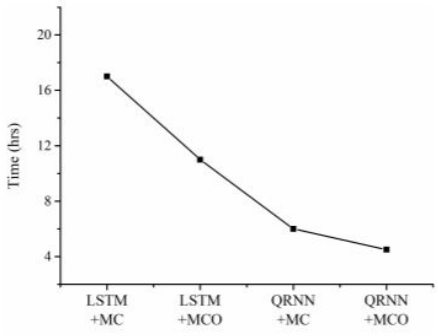

FIGURE 3. **Time consuming of different models.**

Fig 10.0 QuGAN Metrics

Conclusion

In this paper, we propose the QuGAN model which combines QRNN and RL. And the BERT model, optimized reward strategy and Monte Carlo search strategy are used to improve the performance. However, due to not considering the Tibetan grammar characteristics, there are certain invalid questions.

Chapter 3

REPORT ON PROPOSED SYSTEM:

3.1 Problem Statement Analysis:

Problem Statement:

To design and develop an efficient cloud-based system that is capable of providing support and assistance to teaching faculty.

Also to implement Handwritten Text Recognition for student's scanned assignments, Question and Answer Generating System from any given data and Closed Domain Question Answering System for answering any questions from available resources for Teachers Assistance.

The analysis of the problem statement depends on various factors. For the proper analysis, the model needs to be trained with efficient and accurate data. The model needs to be trained in such a manner, that it can effectively produce a quality question and predict the answer to it. Initially, the given PDF’s text has to be extracted and preprocessed into tokens of sentences and fed into the Closed Domain Question Generation Model.

3.2 Design and Methodology of the Proposed System:

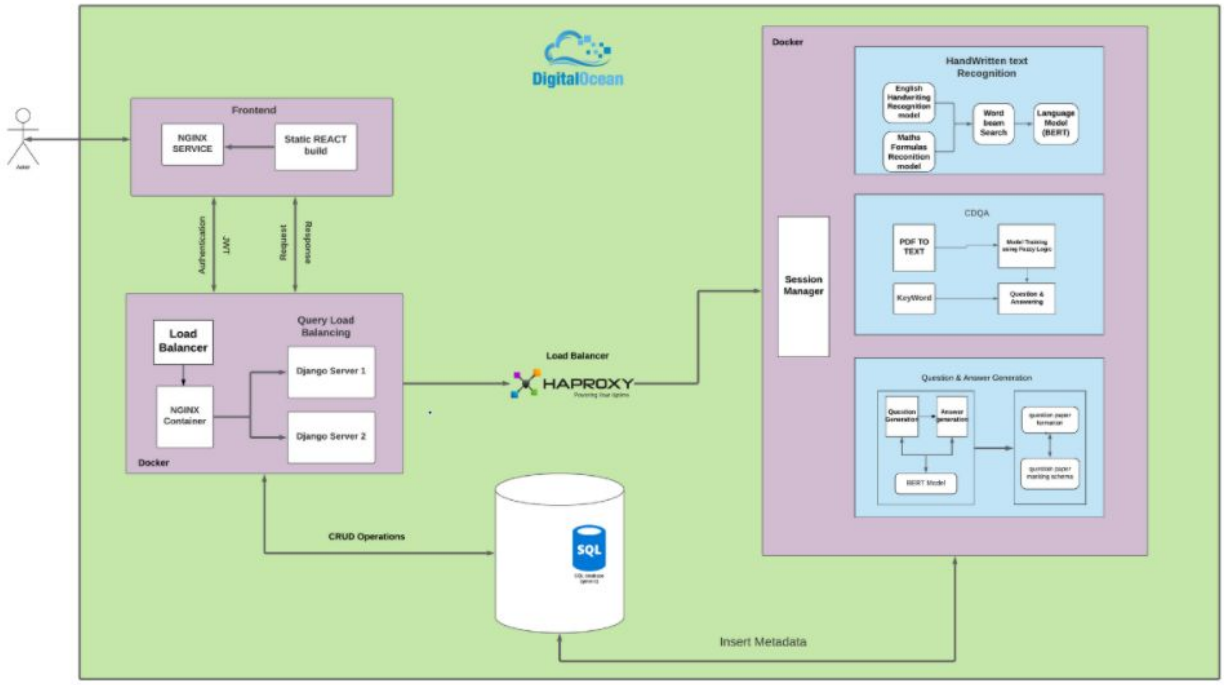

Figure 11: Architecture of proposed system

Methodology:

In the front-end, we have an NGINX Service serving a Static React build of the React-js WebApp. On this web-app, is our Virtual Assistant (VAsst.) deployed. Inputs to this could be a text paragraph or a .pdf/.doc/.docx/.txt file(s). Once any of these input(s) is received at the frontend, it's sent to the backend of our VAsst for further processing.

In order to have a proper Scaled and distributed Architecture, we have split the backend into two, one of which is a Docker container having multiple instances of the Django project, distributing the incoming requests between themselves with the help of an NGINX load balancer, deployed on it. This container serves the react-js web app. All the input(s) from the front-end is trimmed, tokenized, lemmatized and brought into the

input format of the Question Generation/Answering models. Once done with the preprocessing, the data is sent to the other Module of the back-end.

Preprocessing, Loading and running multiple heavy models on the same end could compromise the speed and predictability of the whole system. Hence, leaving no room for the upcoming API hits. So, to get the maximum efficiency out of the system, a distributed/ scalable architecture was necessary. Thus, the other module of the back-end, i.e. is a flask app, where all the heavy models are deployed. The Models deployed here are:

Closed Domain Question Answering/Generation Model :

Multiple datasets for the question generation / answering task have been introduced (S.Q.U.A.D 2.0, NewQA, NarrativeQA, etc). The VAsst. System models are trained and fine tuned on the S.Q.U.A.D 2.0 dataset achieving an accuracy of 82.63% and 89.017%.

For answer aware question generation we usually need 3 models, first which will extract answer like spans, second model will generate question on that answer and third will be a QA model which will take the question and produce an answer, then we can compare the two answers to see if the generated question is correct or not.Having 3 models for single task is lot of complexity, so goal is to create a multi-task model which can do all of these 3 tasks

1.extract answer like spans

2.generate question based on the answer

3.QA

T5 model is fine-tuned in multi-task way using task prefixes

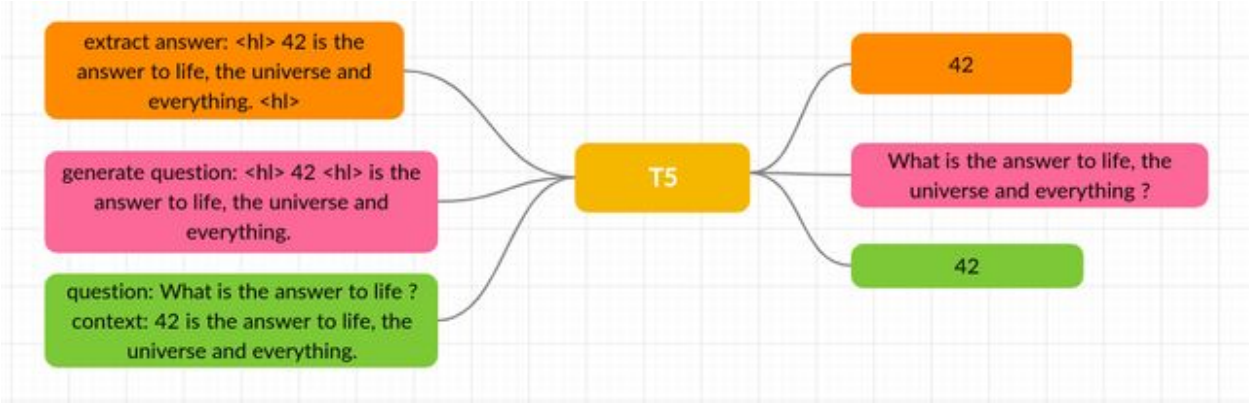

Fig 12.0 Ideology of System

Both of these Modules of the Back-end are connected via another Load balancer i.e. HAProxy. With the help of the efficiently implemented round-robin algorithm in the H.A. Proxy, more than 20k requests can be handled at the same time, making it the perfect fit for the VAsst. System.

3.3 Algorithms Used:

3.3.1 Deep Learning:

Deep learning is a sub-field of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. It is an AI function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the

world. The enormous amount of data is readily accessible and can be shared through fintech applications such as cloud computing. Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks, and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.

3.3.2 Transformers:

The **Transformer** is a deep learning model introduced in 2017, used primarily in the field of natural language processing (NLP). Like recurrent neural networks (RNNs), Transformers are designed to handle sequential data, such as natural language, for tasks such as translation and text summarization.Transformers have become the model of choice for tackling many problems in NLP, replacing older recurrent neural network models such as the long short-term memory (LSTM). Since the Transformer model facilitates more parallelization during training, it has enabled training on larger datasets than was possible before it was introduced.

3.3.3 BERT

BERT is a bi-directional transformer for pre-training over a lot of unlabeled textual data to learn a language representation that can be used to fine-tune for specific machine learning tasks. While BERT outperformed the NLP state-of-the-art on several challenging tasks, its performance improvement could be attributed to the bidirectional

transformer, novel pre-training tasks of Masked Language Model and Next Structure Prediction along with a lot of data and Google's compute power.

3.3.4 Rank BM25 model

BM25 is a ranking function used by search engines to estimate the relevance of documents to a given search query. BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of their proximity within the document. It is a family of scoring functions with slightly different components and parameters.

3.3.5. Hugging face transformers

The Hugging Face transformers package is an immensely popular Python library providing pretrained models that are extraordinarily useful for a variety of natural language processing (NLP) tasks. While the library can be used for many tasks from Natural Language Inference (NLI) to Question-Answering, text classification remains one of the most popular and practical use cases.

Chapter 4

Technologies Used

4.1 Python:

Python is a high-level, object-oriented programming language created by Guido van Rossum. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. Python is a general purpose language having a wide range of applications from Web development, scientific and mathematical computing to desktop graphical user Interfaces. The syntax of the language is clean and the length of the code is relatively short.

4.2 Google Colab Notebook:

Colab is a free cloud source service that enables one to improve his python programming coding skills. It also lets a user develop deep learning applications using popular libraries such as Keras, TensorFlow, PyTorch, and OpenCV. Colab provides GPU and is totally free.

4.3 Jupyter Notebook:

The notebook extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the

whole computation process: developing, documenting and executing code as well as communicating the results.

4.4 Libraries Used

4.4.1 NumPy:

NumPy is a library for the Python programming language, adding

support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

4.4.2 Scikit-Learn:

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, k-means and is designed to incorporate with the Python numerical and scientific libraries NumPy and SciPy

4.4.3 Skimage:

Scikit-image is an image processing package that works with numpy arrays and the package is imported as skimage. It is simple and an efficient tool for image processing and computer vision techniques. It is

accessible to everybody and reusable in various contexts. It is open-source and is built on top of NumPy, SciPy and matplotlib.

4.4.4 Pandas:

Panda is a popular Python-based data analysis toolkit which can be imported using import pandas as pd. It presents a diverse range of utilities, ranging from parsing multiple file formats to converting an entire data table into a NumPy matrix array.

4.4.5 Utils:

Python Utils is a collection of small python functions and classes which make common patterns shorter and easier. This module makes it easy to execute common tasks in python scripts such as converting text to numbers and making sure a string is in Unicode or bytes format.

4.4.6: OpenCV:

OpenCV is a library of programming functions mainly aimed at real-time computer vision. OpenCV-Python makes use of Numpy which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib.

4.4.7 Keras:

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It supports convolutional neural networks and recurrent networks as well as a combination of the two. It allows easy and fast prototyping as it was developed with a focus on enabling fast experimentation with deep neural networks. Keras was initially developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). It offers a high-level, more intuitive set of abstractions that make it easy to develop deep learning models regardless of the computation background used. It contains numerous implementations of neural network building blocks such as layers, objectives, activation-functions and a host of tools to make working with images and text easier.

4.4.8 TensorFlow:

It is a free and open-source software library used for dataflow and differential programming across a range of tasks. It is a symbolic math library used for machine learning applications such as neural networks. Its flexible architecture allows for the easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. TensorFlow computations are expressed as stateful dataflow graphs. The name TensorFlow derives from the operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

TensorFlow programming unit is a programmable AI accelerator designed to provide high throughput of low-precision arithmetic and oriented towards using or running models rather than training them.

Chapter 5

Results and Conclusion

5.1 Results:

```
input_text = "The 2020 Indian Premier League, also known as IPL 13 and
branded as Dream11 Indian Premier League 2020, was the thirteenth
season of the IPL, a professional Twenty20 cricket (T20) league
established by the Board of Control for Cricket in India (BCCI) in
2007. v The tournament was originally scheduled to commence on 29 March
2020, but was suspended until 15 April due to the global coronavirus
pandemic. v After Indian Prime Minister Narendra Modi announced on 14
April that the lockdown in India would last until at least 3 May 2020,
the BCCI suspended the tournament indefinitely. \

On 2 August 2020, it was announced that the tournament would be played
between 19 September and 10 November 2020 in the United Arab Emirates.
\

On 4 August 2020, Vivo pulled out as the title sponsor of the Indian
Premier League (IPL) for this year's edition. \

On 18 August, fantasy cricket league platform Dream11 was named the
title sponsor with a bid of ₹222 crore (US$31 million) for the
tournament. \

Mumbai Indians were the defending champions, and they successfully
defended their title following a 5 wicket win over Delhi Capitals in
the final on 10 November, 2020."
```

Figure 13: Paragraph Input

```
What was the 2020 Indian Premier League branded as?
When did Narendra Modi announce that the lockdown in India would last until at least 3 May 2020?
When was the 2020 Indian Premier League announced?
Who pulled out as the title sponsor of the Indian Premier League in 2020?
Which fantasy cricket league platform was named the title sponsor of the IPL?
How many wickets did Mumbai Indians win over Delhi Capitals in the final?
```

Figure 14: Questions Generated

```
Dream11
14 April
2 August 2020
Vivo
Dream11
5 wicket
```

Figure 15: Answers Generated

```
[{'answer': 'Dream11',
  'question': 'What was the 2020 Indian Premier League branded as?'},
 {'answer': '14 April',
  'question': 'When did Narendra Modi announce that the lockdown in India would last until at least 3 May 2020?'},
 {'answer': '2 August 2020',
  'question': 'When was the 2020 Indian Premier League announced?'},
 {'answer': 'Vivo',
  'question': 'Who pulled out as the title sponsor of the Indian Premier League in 2020?'},
 {'answer': 'Dream11',
  'question': 'Which fantasy cricket league platform was named the title sponsor of the IPL?'},
 {'answer': '5 wicket',
  'question': 'How many wickets did Mumbai Indians win over Delhi Capitals in the final?'},
```

Figure 16: Question Answer Pair

Description :

First Pdf will be taken as input from the user. Then, the text will be extracted from the pdf. This text will be sent as an input for our model with the option of what task is needed to be performed by the user. After selection of the task to be performed, the text of the pdf will be processed by our respective model and the required result will be given as the output.

5.2 Conclusion:

This Virtual Assistant could be used in various educational as well as industrial institutes that will help them increase the productivity level and also help the users to better understand the concepts contained in the file and especially for the educational sector for the students and the teachers. With the help of the complex architecture of the model used, we are able to achieve an accuracy of 82.631% on the question generation model and 89.017% on answer generation model.

Chapter 6

References

1. A. Agarwal, N. Sachdeva, R.K Yadav, V.Udandarao, V.Mittal, "EDUQA: Educational Domain Question Answering System Using Conceptual Network Mapping", SBILab, Department of ECE, IIIT-Delhi, India, Million Sparks Foundation, ICASSP, 978-1-5386-4658-8 2019 IEEE
2. Darshana V Vekariya, Nivid R Limbasiya, "A Novel Approach for Semantic Similarity Measurement for High Quality Answer Selection in Question Answering using Deep Learning Methods", 2020 6th International Conference on Advanced Computing & Communication Systems (ICACCS), 978-1-7281-5197-7 2020 IEEE.
3. Vignesh A, Monisha Devi, Venkateshwaran G K, Hariharan K " Semantics Enhanced Answer Selection in Closed-domain Question Answering System", 2018 International Conference on Power, Information and Communication (ICCPEIC), 978-1-5386-2447-0 2018 IEEE
4. Emanuele Damiano, Raffaele Spinelli, Massimo Esposito, Giuseppe De Pietro, "Semantics-Enhanced Answer Selection in Closed-domain Question Answering System", 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems. DOI 10.1109/SITIS.2016 IEEE
5. Marco Pota, Massimo Esposito, Giuseppe De Pietro, "Learning to Rank Answers to Closed-Domain Questions by using Fuzzy Logic", Natural Research Council of Italy. 978-1-5090-6034-4 2017 IEEE.
6. Sweta P. Lende, Dr.M.M. Raghuwanshi “Question Answering System on Education Acts Using NLP Techniques” in 2016 EEE Sponsored World

Conference on Futuristic Trends in Research and Innovation for Social Welfare (WCFTR'16)

7. Ajitkumar Meshram Pundge, C. Namrata Mahender " Evaluating Reasoning in Factoid based Question Answering System by Using Machine Learning Approach in Proceedings of the International Conference on Communication and Electronics Systems (ICCES 2018), IEEE Xplore Part Number:CFP18AWO-ART; ISBN:978-1-5386-4765-3.
8. Animesh Srivastava, Shantanu Shinde, Naeem Patel, Siddhesh Despande, Anuj Dalvi, Shweta Tripathi " Questionator - Automated Question Generation using Deep Learning" in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE).
9. Akshay Upadhya B, Sowmya Kamath S, Swastik Udupa " Deep Neural Network Models for Question Classification in Community Question-Answering Forums" in 10th ICCCNT 2019 July 6-8, 2019, IIT - Kanpur, IEEE 45670.
10. Yuan Sun, Chaofan Chen, Tianci Xia and Xiaobing Zhao, "QuGAN: Quasi Generative Adversarial Network for Tibetan Question Answering Corpus Generation", Volume 7, IEEE 2019