**K. K. Wagh Institute of Engineering Education & Research, Nashik**

Department of Information Technology

**(Academic Year:2020-21)**

**Project Title: Text Matching for Question-Answer System**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Group No.: 01** | | |  | **Guide Name : Prof. R.M. Bora** |  |  |
|  |  |  |  |  |  |  |
|  | **GROUP MEMBERS:** | |  |  |  |  |
|  | **Roll No./** | **Name of Student** |  | **Project Area** | **Project Platform** |  |
|  | **Seat No.** |  |  |
|  |  |  |  |  |  |
|  | 22 | Gangurde Srushti |  |  |  |  |
|  | 30 | Kanchwala Abdulhussain |  | Machine Learning, | Django Framework |  |
|  | 32 | Karlekar Anuja |  | Deep Learning, Natural Language Processing |  |  |
|  | 35 | Kothawade Yashodhan |  |  |  |  |
|  | 36 | Kulkarni Amogh |  |  |
|  |  |  |  |  |  |  |

1. **Abstract**

For an automated Question-Answer Matching System, most important tasks are: to accurately map keywords in the model answer to user’s answer, to check the accuracy in text semantics, and evaluate the answer. This project of Q&A system implements deep semantic matching and extraction by matching information from different angles. It uses Multiple-Perspective Semantic Crossover (MPSC) model for modeling semantic-based match of questions and answers. Inputs to the MPSC model are text files including: Model Question set, Model Answer set, Marks Distribution provided by Administrator and Answer Sheet provided by Student. The model is responsible for calculating matching degree between questions and probable answers. On the basis of this matching degree, system generates output that is an approximate score based on syntactic, semantic and pragmatic relevance of the answer. The MPSC model generates a compact semantic crossover by referring to the text, eluding the limitations of word vector and vector matrices. It integrates Neural Network Models like LSTM and CNN for semantic analysis and accurate feature extraction from the text. This project also uses Binary Cross Entropy as an objective loss function for identifying duplicate questions or answer sentences.

**Keywords:** Question-Answer Matching, Neural Network, LSTM, CNN, Binary Cross Entropy, Adadelta optimizer.

**B) Objectives**

* To build a working model of question and answer matching.
* To accelerate the paper assessment and reduce workload on examiner.
* To provide unbiased paper evaluation.

1. **Scope**

To implement an optimize Question Answer matching system having a better accuracy and a higher degree of similarity. The Key features are:

* Text semantic analysis
* Text feature extraction
* Matching degree calculation
* Scoring/Grading of answer

**D) Literature Survey**

**B. Hu, Z. Lu, H. Li, and Q. Chen. "Convolutional neural network architectures for matching natural language sentences." In Advances in Neural Information Processing Systems, 2014.**

**ARC-I:**

Architecture-I (ARC-I), takes a conventional approach: It first finds the representation of each sentence, and then compares the representation for the two sentences with a multi-layer perceptron (MLP).It is based on a single-text that is trying to represent the text with a vector and then calculating the similarity of the text vector. Although ARC-I enjoys the flexibility brought by the convolutional sentence model, it suffers from a drawback inherited from the Siamese architecture: it defers the interaction between two sentences, therefore runs at the risk of losing details (e.g., a city name) important for the matching task in representing the sentences. In other words, in the forward phase (prediction), the representation of each sentence is formed without knowledge of each other. This cannot be adequately circumvented in backward phase (learning), when the convolutional model learns to extract structures informative for matching on a population level.

**ARC-II:**

In view of the drawback of Architecture-I, we propose Architecture II (ARC-II) that is built directly on the interaction space between two sentences. It has the desirable property of letting two sentences meet before their own high-level representations mature, while still retaining the space for the individual development of abstraction of each sentence. ACR-II performs modelling directly and it considers that the two texts can be interacted earlier, and extracting deep interactive information is more conductive to solve text matching problems. Basically, in Layer-1, we take sliding windows on both sentences, and model all the possible combinations of them through “one-dimensional” (1D) convolutions. Clearly the 1D convolution preserves the location information about both segments. After that in Layer-2, it performs a 2D max-pooling in non-overlapping 2 X 2 windows. In Layer-3, we perform a 2D convolution on k3 X k3 windows of output from Layer-2. This could go on for more layers of 2D convolution and 2D max-pooling, analogous to that of convolutional architecture for image input.

**Liang Pang Yanyan Lan Jiafeng Guo Jun Xu Xueqi Cheng. Manning. 2018.Stanza: "A Study of MatchPyramid Models on Ad-hoc Retrieval"**

**MatchPyramid:**

A recently introduced deep matching model, namely MatchPyramid, which has shown state-of-the-art performances on several text matching tasks. In MatchPyramid, local interactions between two texts are first built based on some basic representations (e.g., word embeddings). The local interactions represented by a matching matrix is then viewed as an image, and a convolutional neural network (CNN) is employed to learn hierarchical matching patterns. Finally, the high-level matching patterns are fed into a multilayer perceptron to produce the matching score between the two texts. The three parts MatchPyramid are as follows:

1. Matching Matrix:

Matching Matrix is a two-dimension structure where each element Mij denotes the similarity between the i-th word wi in the first piece of text and the j-th.

2. Hierarchical Convolution:

Based on the Matching Matrix, MatchPyramid conducts hierarchical convolution to extract matching patterns. Hierarchical convolution consists of convolutional layers and dynamic pooling layers, which are commonly used in CNN (such as AlexNet, GoogLeNet) for image recognition tasks.

3. Matching Score Aggregation:

After hierarchical convolution, 6 two additional fully connected layers are used to aggregate the information into a single matching score.

**E) Deliverables**

* Web Application with Admin and Student Login
* Build Machine Learning Model ( .h5 file)

**F) Resource Requirements (Hardware/Software etc.).**

* Hardware
  + GPU
* Software
  + Python3
  + OS: Window 8 and above, Linux
  + Internet connectivity