**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 the 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 the 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**

For the task of question answer matching, semantics analysis and text matching is quite necessary. These processes of semantic analysis and text matching have been previously implemented through various models such as the ARC-I, ARC-II and MatchPyramid. These machine learning models are based on text vectorization and processing of actual text with single direction and are highly dependent on vector length to retain information. However, all these models fail to retain information as the vector size increases. To overcome these limitations of the traditional models the MPSC combines the LSTM and CNN models where LSTM retains the large vector data in multiple directions and CNN generates model based on these matrices.

**ARC-I:**

ARC-I [1] i.e Architecture-I , makes use of a conventional approach for sentence matching: firstly the representation of each sentence is found out , and then comparison of the representation for the two sentences is done using a multi-layer perceptron (MLP). ARC-I bases on a single-text, it represents the text with a vector and then calculates the similarity of the text vector. Although there is flexibility due to the convolutional sentence model, there occurs a drawback carried on from Siamese architecture to the ARC-I, the architecture defers the interaction between two sentences, therefore there is a risk of losing details which are necessary for the matching task in representing the sentences. This means, in the forward phase (prediction), each sentence representation is formed without knowledge of another. This becomes difficult to circumvent in the backward phase (learning), when the convolutional model learns to extract structures informative for matching on a population level.

**ARC-II:**

To overcome the drawback of Architecture-I, Architecture II (ARC-II) [1] was proposed. ARC-II is built directly on the interaction space between two sentences. It has the property of allowing two sentences to meet before their own high-level representations are formed, although even while doing this it retains the space for the individual development of abstraction of each sentence. In ACR-II modelling occurs directly by considering that the two texts can be interacted earlier, and that extraction of deep interactive information can be more conductive to solve text matching problems. In Layer-1, there are sliding windows on both sentences, and modelling of all the possible combinations of these through “one-dimensional” (1D) convolutions. Evidently the 1D convolution maintains the location information about both segments. In Layer-2, there is a 2D max-pooling in non-overlapping 2 X 2 windows. In Layer-3, a 2D convolution on k3 X k3 windows of output from Layer-2 is done. This may keep going for more layers of 2D convolution and 2D max-pooling, similar to that of convolutional architecture for image input.

**MatchPyramid:**

MatchPyramid [2] is a deep matching model, a model which on several text matching tasks has shown exceptional performances. MatchPyramid builds local interactions between two texts based on some basic representations like word embeddings. These local interactions which are represented using a matching matrix are then considered as an image, which are then fed to the convolutional neural network (CNN) to learn hierarchical matching patterns in them. After all this, the matching score between the two texts are generated by a multilayer perceptron using the high-level matching patterns. The three parts MatchPyramid are as follows:

1. Matching Matrix:

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

2. Hierarchical Convolution:

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

3. Matching Score Aggregation:

Once the hierarchical convolution is done, 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

**G) References**

1. 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.
2. Liang Pang Yanyan Lan Jiafeng Guo Jun Xu Xueqi Cheng. Manning. 2018.Stanza: "A Study of MatchPyramid Models on Ad-hoc Retrieval"