Project Abstract

In today’s world people use the internet to get answers to a wide range of questions. A person in some part of the world can ask a question and get an answer from someone present in some other geographical location. However, in our IIIT campus we mostly confine our technical discussion and knowledge sharing to our small group of friends. Questions posted on moodle are mostly directed towards the TAs or the professor of a course. We feel that there is a need to have a platform in the institute where technical questions can be posted by any student and solutions can be discussed and posted by fellow peers.

Solution Overview

Web Application:

We would be building a web application that basically provides two main purposes:

(a) **Posting a new question raised by the user:**
The user will be displayed a form to enter the relevant details regarding the question to be posted by the user, such as category of the question, visibility, etc.

(b) **Search the questions relevant to the query asked by the user:**
The user can see the various questions posted by the other users that are relevant to the query done by the user. The relevant results will be shown with the help of the tags and the categories of the questions posted.

The user could be a student or a professor.

Technologies to be Used:

Python Flask as backend, HTML/ CSS/ JavaScript for the frontend and SQLite as the Database.

ML Application :

Whenever a user query, this platform should check all questions which are relevant to this query. So we try to understand the content of what the user is trying to search for, and then return the most similar results based on that.
This can be done using tag prediction and similar questions retrieval.

This can be done as follows -

1. Train a Tag Classifier using the LSTM model.
2. Collect all the questions with a similar tag (query's tag).
3. After building tag classifier we measure the semantic similarity between users' query and all the questions using cosine similarity.
   We can also take into account the popularity of the question and responses that people have made by adding a score (composed of number of votes on the question and answers) in cosine distance.

For building Tag classifier we will select top 600-700 tags. We will use stackoverflow data and then train our classifier (LSTM Model).

To measure the semantic similarity we will use **fasttext** embeddings. After creating sentence embeddings, we will measure distance between both the embeddings using cosine distance and rank by the obtained distance.

Distributed System Application:

Each node in the distributed network would have an ML model running on it to process the results parallely. The questions will be stored in the various nodes. Whenever the user requests for the relevant questions related to the input tags/words, each node checks for the similarity in the questions for which that node is responsible. So using a master node, we can compile all other results obtained and show them to the user in a collective manner.
All the searching for each node occurs parallely, so as to fasten up the search results.

Our proposed model should be distributed:

1. To maintain and distribute the load fairly.
   This ensures that a server is not overloaded.
2. so that even if there is a hardware failure, we should have multiple copies of the same server in the system.
3. And finally, we must ensure that the requests are responded to in a timely fashion.