Team 1

Austin Fishbaugh, Bindu Kumari, Daniel Lamkin, Kaixin Zhang

# Project Proposal

**1. Task: What task are you solving?**

**a. Please give a clear definition! If possible define variables**

1. Get all the questions related to a certain topic that has no answer. (Lots of view, but no answer.)

2. List some of the most relevant questions without answer to user's query, and someone knowledgeable in the topic can provide answers to questions which have the most popularity but no answers.

**b. What are the inputs? What kind of data do you need?**

Input: Questions from user as query.

Data needed: All questions in Stack Overflow.

**c. What is your outputs?**

List out the most relevant question without answer related to user's query, ranked by most viewed to least viewed

(Notice, a task only states WHAT is to be solved, but not HOW to solve it!)

**2. Motivation: Why?**

**a. In which domain are you solving this task?**

Education

**b. Why is it important to solve this task?**

To improve the efficiency to locate the most wanted un-answered questions.

To improve the accuracy for user to find what he/she wants for an answer.

**c. How is this helping a user?**

This is helping the user because if someone knowledgeable in the topic can find the questions which are viewed by many users but do not have a good answer, they can answer these questions first, as they have the most amount of users waiting for the answer. This means user is more likely to get an answer upon first searching for a query, and it answers the question for all those who were waiting for the answer.

**3. Method: How?**

**a. How are you solving this task?**

Index all questions from Stack Overflow

Take a query from user

Rank all relevant docs

Use Language Models to go through content of a question to find most likely related to the query.

Rank the relevant documents first by number of answers, then by number of views. Least answers with most views would be at top.

**b. Which algorithms are you applying?**

Use TFIDF to rank relevant

Language Model

(Maybe applied rank function from Programming assignment 5)

Ranking of documents by answers and views is simple comparison of numerical values

**4. Evaluation: Success?**

**a. Which data are you using?**

Questions data from Stack Overflow (Based on Stack's own API)

**b. What is the ground truth (e.g. how do you know what is correct/relevant?)**

Come up with different user cases, and then compared the result of this application with the result of google & build-in search function of Stack Overflow

**c. Which evaluation measures do you plan on using?**

We will evaluate using the standard measure from the class, NDCG, MAP, and Precision@R.

5. **Expectation**

**a. What do you expect to come out of this?**

An end-to-end application that can provide helpful functionality to those trying to answer questions on stack overflow. Allows a user knowledgeable in a topic to easily find questions on that topic that are in most need of an answer.

**Project structure:**

BackEnd:

- Core:
  - Index
  - Main Search Functions
  - Ranking functions
- Agent:
  - GET API: send GET request to Stack Overflow to retrieve all questions
  - GET API client: Receive GET request with parameter(Query String) from UI, then return result.

FrontEnd:

- Html Page: Input search query and display results.

**Possible implantations:**

Backend: Lucene, ranklib(Prog5), Jersey(RESTful APIs), Gson(JSON String formatter)

Frontend: HTML, css, JavaScript, Jquery.