

CS 453, Fundamentals of Information Retrieval, Winter 2015

Project Assignment 2

Suggesting Queries Based on Word Similarity and Query Modification Patterns

due Thursday, February 19

Project Outline

This project assignment consists of implementing the *query-suggestion* approach, *WebQS*, presented in the paper, “Assisting Web Search Using Query Suggestion Based on Word Similarity Measure and Query Modification Patterns,” published in the *Journal of World Wide Web*, Volume 17, Number 5, 2014. You must implement *WebQS* proposed in the paper to generate suggestions for a user’s query.

1 Project Description

WebQS provides a guide to the users for formulating/completing a keyword query Q using suggested keywords (extracted from the AOL query logs) as potential keywords in Q . The query-suggestion approach considers *initial and modified queries* in the AOL query logs, along with *word-similarity measures*, in making query suggestions. *WebQS* facilitates the formulation of queries in a *trie* data structure and determines the *rankings* of suggested keyword queries using distinguished features exhibited in the raw data in the AOL query logs.

1.1 The AOL Query Logs

WebQS relies on the AOL query logs to suggest queries. The logs of AOL, which include 50 million queries that were created by millions of AOL users over a three-month period between March 1, 2006 and May 31, 2006. The query logs are made available to the class for this project assignment and posted under the CS 453 Project homepage.

An AOL query log includes a number of query sessions, each of which captures a period of sustained user activities on the search engine. Each AOL session differs in length and includes a (i) user ID, (ii) the query text, (iii) date and time of search, and (iv) optionally clicked documents. A *user ID*, which is an anonymous identifier of its user who performs the search, determines the boundary of each session (as each user ID is associated with a distinct session). *Query text* are keywords in a user query and multiple queries may be created under the same session. The *date* and *time* of a search can be used to determine whether two or more queries were created by the same user within 10 minutes, which is the time period that dictates whether two queries should be treated as *related*. *Clicked documents* are retrieved documents that the user has clicked on and are ranked by the search engine. Queries and documents include *stopwords*, which are commonly-occurring keywords, such as prepositions, articles, and pronouns, that carry little meaning and often do not represent the content of a document. Stopwords are not considered by *WebQS* during the query creation process.

You are supposed to implement *WebQS* by parsing the AOL query logs to extract query keywords while at the same time retains the information of *related* keywords in the same session, which were submitted by the same user within 10 minutes in the same session, as discussed earlier.

1.2 The Trie Data Structure

Using the extracted keywords, *WebQS* constructs a trie T in which each node is labeled by a *letter* in an extracted keyword in the given order, and each node in T is categorized as either “complete” or “incomplete.” A *complete* node is the last node of a path in T representing an (a sequence of,

respectively) extracted query keyword (keywords, respectively). If node c is a complete node, then T_c (the subtree of T rooted at a child node of c) contains other suggested keyword(s) represented by the nodes in the path(s) leading from, and excluding, c . The possible number of suggestions of a (sequence of) keyword(s) K rooted at T_c is n , where n is the number of complete nodes in subtrees rooted at T_c , and K is the (sequence of) keyword(s) extracted from the root of T_c . An *incomplete* node is the last node of a path P in T such that P does not yield a (sequence of) word(s). If c is an incomplete node, then all subsequent nodes of c up till the first complete node are potential suggestions of keywords represented by the nodes in the path leading from, and including, c .

WebQS retains the keywords in query texts in a *trie* data structure using queries in the AOL query logs. Using the trie, candidate keywords suggested for a query can be found and ranked dynamically. To suggest potential query keywords, WebQS locates a trie branch b up till the (letters in the) keywords that have been entered during the query creation process and extracts the subtrees rooted at the child nodes of the last node of b . The extracted suggestions are ranked using a set of features (presented in Section 1.3).

To simplify this assignment, you can assume that any initial query Q entered by the user is spelled correctly.

1.3 Ranking Possible Suggestions

WebQS ranks suggested query keywords in its trie data structure based on (i) the *frequency of occurrence* (*freq*) of the keywords in the AOL query logs, (ii) their *similarity* with the keywords submitted by a user based on the word-correlation factors (*WCFs*), and (iii) the *number of times* the keywords in user queries were *modified* (*Mod*) to the keywords in the suggested queries within 10 minutes as shown in the query logs. (For this project assignment, instructions will be given to you which show you how to access the word-correlation matrix.)

Given that SQ is a suggested query for a(n) (in)complete user query Q which has been entered during the query construction process, WebQS computes a *ranking* score for SQ , denoted *SuggRank* (Q, SQ), which reflects the degree of *closeness* of SQ to the letters/keywords in Q .

$$SuggRank(Q, SQ) = \frac{freq(SQ) + WCF(Q, SQ) + Mod(Q, SQ)}{1 - Min\{freq(SQ), WCF(Q, SQ), Mod(Q, SQ)\}}$$

where

- $freq(SQ)$ is the *frequency of occurrence* of SQ in the AOL query logs.
- $WCF(Q, SQ)$, which is equal to $WCF(SQ, Q)$, is as defined in Equation 1 in the paper.
- $Mod(Q, SQ)$ is the *number of times* Q is *modified* to SQ in the same session in the AOL query logs within 10 minutes.
- As $freq(SQ)$, $WCF(Q, SQ)$, and $Mod(Q, SQ)$ are in different numerical scales, prior to computing the *SuggRank* of SQ with respect to Q , they are *normalized* using a logarithmic scale to be in the same range.

For this project assignment, you are supposed to offer the top 8 suggestions, which follows the eight results displayed by major web search engines in response, to a user query.

2 Pass-off

To receive credit for this project assignment, you must pass off your program during the TA office hours by the due date.

1. When you pass off your program, you should copy all the files (include your executable file) to the hard disk of a laptop.
2. The TA will try out different queries using different single and multiple keywords at that time, and your program is supported to create suggested queries as the TA enters a sequence of (one or more) keywords.
3. Your suggested queries for each test query Q will be compared with the targeted suggestions of Q generated by WebQS.

The assignment is worth 150 points.