SUITS: Faceted User Interface for Constructing Structured Queries from Keywords

Elena Demidova¹, Xuan Zhou², Gideon Zenz¹, Wolfgang Nejdl¹

¹L3S Research Center, Hanover, Germany  
{demidova, zenz, nejdl}@L3S.de  
²CSIRO ICT Centre, Australia  
 xuan.zhou@CSIRO.au

Abstract. Keyword search and database query are two ways for retrieving data in real world settings. In this demonstration, we show SUITS, a novel search interface over relational databases that smoothly integrates the flexibility of keyword search and the precision of database queries. SUITS allows users to start with arbitrary keyword queries, refine them incrementally by following the suggestions given by the system and finally obtain desired structured queries.

Keywords: keyword search, query construction.

1 Introduction

The digital information accessible today, such as that on the Web, possesses semantics in both structured and unstructured forms. Keyword search, which was originally developed for retrieving documents, is an intuitive and convenient interface for accessing unstructured data. However, keyword search leaves users with limited expressiveness in describing their information needs. As a consequence, users may fail to find desired information. On the contrary, database queries enable users to exploit available structures to achieve more precise queries and corresponding result sets. However, using a database system requires adequate knowledge of the database schema and proficiency in the query language, making data access a difficult task for unskilled users. In this demo, we present SUITS, a novel search interface which provides a layer of abstraction on top of relational databases to smoothly integrate the intuitiveness of keyword search and the expressiveness of database queries.

As shown in Fig. 1, the SUITS interface consists of four parts: a search field for the user to input keyword queries, a result window to present search results (at the bottom), a query window to present structured queries (on the left) and a faceted query construction panel providing query construction options (on the right). To perform search, the user first issues a keyword query, for instance “Fuzz Wright London” intended to search for the movie called “Hot Fuzz” and directed by Edgar Wright. Besides returning a ranked list of results like standard keyword search [1, 2],

* This work is partially supported by the FP7 EU Project OKKAM (contract no. ICT-215032) and TENCompetence Integrated Project (contract no. 027087).
SUITS suggests to the user a list of structured queries in the query window to clarify her intent. The suggested queries assign different semantics to the keywords. Some queries search for the movies with actor “Wright”, while others search for the actors who acted as a character named “London”. If the user identifies the structured query that represents her intent, she can click on it, such that the result window will show the results of that particular query. Otherwise, she can go to the faceted query construction panel to incrementally construct the intended query. The panel lists a number of query construction options suggested by the system. As shown in Fig. 1, the user specifies that “Fuzz” must appear in the movie title and “Wright” must be a director’s name. The query window changes accordingly to zoom into the queries satisfying the chosen options. Such interaction continues iteratively until the user obtains the right query and/or satisfactory search results.

With the SUITS interface, users can start with arbitrary keywords and structure their queries progressively by following the system’s suggestions. Finally, they can either select a completely structured query from the query window or a partially structured query by specifying appropriate query construction options. It depends on the degree to which users want / are able to clarify their search intents.

2 System Description

The query processing of SUITS can be split into two phases: an offline pre-computing phase and an online query phase. In the first phase, SUITS creates inverted indexes for all text columns in the database. It also generates query templates that are potentially employed by users when forming structured queries. A query template is a structural pattern used to query a database. For example, users sometimes search for...
the movies with a certain character, and sometimes search for the actors who have played in a certain movie. Both are commonly used query templates.

The online query phase consists of three steps. In Step 1, upon receiving a user’s keyword query, the system checks the full-text indices for occurrences of the query terms. In Step 2, it combines the term occurrences and the pre-computed query templates to create meaningful structured queries and query construction options. In Step 3, the system ranks the queries according to their likelihood of being intended by the user, and selects a set of query construction options that can more effectively clarify the user’s intent. Finally, it returns the top-k queries and a set of selected options to the user. If the user chooses some of the query construction options, these options are fed back to Step 2 in order to filter out queries and query construction options that do not satisfy the user’s specification.

The success of SUITS relies on the proper ranking of structured queries, effective selection of the query construction options and efficient query processing. In [3], we presented a set of techniques to accomplish these tasks. In this paper, due to the size limitations, we focus only on the query construction.

3. Query Construction

SUITS suggests appropriate query construction options to support incrementally creating the user intended structured query from keywords. Choosing a query construction option is equivalent to specifying a part of the structured query. Thus, query construction options are actually partial queries. As shown in Fig. 3, the complete queries and partial queries constructed from a set of keywords can be organized in a hierarchy. At the bottom of this hierarchy are the smallest partial queries composed of only one keyword and one attribute. In the middle are partial queries that join two or more keywords together. At the top, complete structured queries involving all keywords are located.

During the query construction, the system first presents a small set of partial queries to the user. If the user chooses any of the partial queries, she actually indicates that her intended complete query will contain that partial query. Therefore, the system
can remove from the query hierarchy all the complete queries not containing that partial query. Later on, the system presents another set of partial queries to the user for selection. This process carries on until the user identifies the desired query in the query window.

![Hierarchy of Partial Queries](image)

**Fig. 3.** Hierarchy of Partial Queries

Selecting proper partial queries is crucial in the query construction process. A good strategy allows a user to exclude as many complete queries as possible in each round, so that she can obtain the desired query quickly. In [3], we proposed a number of principles for selection and ranking of the query construction options.

## 4 Demonstration Overview

In this demonstration we will primarily show how SUITS works and how a user can employ it to efficiently identify desired information in a database without any knowledge of the database schema. First, we will demonstrate the complete query process. This process starts from submitting a keyword query to the system, followed by a presentation of the top-k structured queries that give different interpretations of the keywords, followed by an execution of the queries to retrieve search results. Then we show how the query construction options suggested by SUITS can guide users to quickly construct desired structured queries. We also present the result navigation component of SUITS which enables extending a search result to explore the database context. Our demonstration uses a real-world dataset, a crawl of the Internet Movie Database (IMDB).

## References