Abstract

The prevalence of graph-structured data in modern real-world applications has led to a rejuvenation of research on graph data management and analytics. Several database query languages have been proposed for textually querying graph databases. Unfortunately, formulating a graph query using any of these query languages often demands considerable cognitive effort and requires “programming” skill at least similar to programming in SQL. Yet, in a wide spectrum of graph applications consumers need to query graph data but are not proficient query writers. Hence, it is important to devise intuitive techniques that can alleviate the burden of query formulation and thus increase the usability of graph databases. In this dissertation, we take the first step to study the graph query autocompletion problem. We provide techniques that take a user’s graph query as input and generate top-\(k\) query suggestions as output, to help to alleviate the verbose and error-prone graph query formulation process in a visual environment.

Firstly, we study visual query autocompletion for graph databases. Techniques for query autocompletion have been proposed for web search and XML search. However, a corresponding capability for graph query engine is in its infancy. We propose a novel framework for graph query autocompletion (called AUTOG). The novelties of AUTOG are as follows: First, we formalize query composition that specifies how query suggestions are formed. Second, we propose to increment a query with the logical units called \(c\)-prime features, that are (i) frequent subgraphs and (ii) constructed from smaller \(c\)-prime features in no more than \(c\) ways. Third, we propose algorithms to rank candidate suggestions. Fourth, we propose a novel index called feature DAG (FDAG) to further optimize the ranking.
Secondly, we propose user focus-based graph query autocompletion. AUTOG provides suggestions that are formed by adding subgraph increments to arbitrary places of an existing user query. However, humans can only interact with a small number of recent software artifacts in hand. Hence, many such suggestions could be irrelevant. We present the GFOCUS framework that exploits a novel notion of user focus of graph query formulation. Intuitively, the focus is the subgraph that a user is working on. We formulate locality principles to automatically identify and maintain the focus. We propose novel monotone submodular ranking functions for generating popular and comprehensive query suggestions only at the focus. We propose efficient algorithms and an index for ranking the suggestions.

Thirdly, we propose graph query autocompletion for large graphs. Graph features that have been exploited in AUTOG are either absent or rare in large graphs. To address this, we present Flexible graph query autocompletion for LArge Graphs, called FLAG. We propose wildcard label for query graph and query suggestions. In particular, FLAG allows augmenting users’ queries using subgraph increments with wildcard labels, which summarize query suggestions that have similar increment structures but different labels. We propose an efficient ranking algorithm and a novel index, called Suggestion Summarization DAG (SSDAG), to optimize the online suggestion ranking.

Detailed problem analysis and extensive experimental studies consistently demonstrate the effectiveness and robustness of our proposed techniques in a broad range of settings.

Keywords: Subgraph query, query autocompletion, graphs, database usability.
# Table of Contents

Declaration i  
Abstract ii  
Acknowledgements iv  
Table of Contents v  
List of Tables ix  
List of Figures xi  

## Chapter 1 Introduction  
1.1 Visual Query Autocompletion for Graph Databases 2  
1.2 User Focus-based Graph Query Autocompletion 6  
1.3 Graph Query Autocompletion for Large Graphs 11  
1.4 Summary 14  

## Chapter 2 Literature Review  

## Chapter 3 AutoG: A Visual Query Autocompletion Framework for Graph Databases  
3.1 Preliminaries 19  
3.1.1 Subgraph Queries and Background 19  
3.1.2 Query Composition 20  
3.1.3 Query Composition Modes 21
3.2 c-Prime Features .................................................. 23
  3.2.1 Properties of c-Prime Features .......................... 27

3.3 Autocompletion Framework for Subgraph Queries – AutoG .......... 28
  3.3.1 Query Decomposition .................................. 29
  3.3.2 Generation of Candidate Suggestions .................... 31
  3.3.3 Ranking Candidate Suggestions ........................ 32
  3.3.4 Analysis of the Ranked Subgraph Query Suggestion (RSQ) Problem 41

3.4 Indexed Autocompletion for Subgraph Queries – AutoGI .......... 42
  3.4.1 Feature DAG (FDAG) Index ............................. 43
  3.4.2 Autocompletion Using FDAG ............................ 47
  3.4.3 The FDAG Construction ................................. 47

3.5 Pruning Redundant Compositions via Graph Automorphism .......... 50

3.6 Experimental Evaluation ...................................... 52
  3.6.1 Suggestion Quality ...................................... 54
  3.6.2 Index Construction Performance .......................... 57
  3.6.3 Online Autocomplete Performance ....................... 60
  3.6.4 Suggestion Qualities with Different Underlying Definitions .... 62
  3.6.5 Online Performance Breakdowns ........................ 65

3.7 Summary .......................................................... 68

Chapter 4  GFocus: User Focus-based Graph Query Autocompletion 74

4.1 Background on Graph Query Autocompletion (gQAC) ............ 74

4.2 Query Formulation Operators .................................. 78

4.3 Locality Principles for Query Formulation ........................ 80
  4.3.1 Temporal Locality Principle ............................. 81
  4.3.2 Structural Locality Principle ............................. 82
  4.3.3 Decay-and-Propagation Algorithm ........................ 83
  4.3.4 The Characteristics of User Attention Weights ............ 84

4.4 User Focus of GQAC ............................................. 85
  4.4.1 The Pseudocode for Determining the User Focus ........... 87
Chapter 4  Towards Graph Query Autocompletion for Large Graphs  123

4.5 GQAC at User Focus ............................................... 87
  4.5.1 Candidate Suggestions at User Focus ....................... 88
  4.5.2 Ranking Suggestions ......................................... 92
4.6 Indexed GQAC at User Focus ...................................... 95
  4.6.1 Structural-Union-of-Suggestions DAG ....................... 96
  4.6.2 Autocompletion Using SUDAG ............................... 97
  4.6.3 The SUDAG Index Construction .............................. 99
4.7 Experimental Evaluation .......................................... 99
  4.7.1 Suggestion Quality via User Study .......................... 104
  4.7.2 Suggestion Quality via Simulations ......................... 107
  4.7.3 Online Autocompletion Performance ....................... 112
  4.7.4 Additional Experiments .............................. 115
4.8 Summary ......................................................... 118

Chapter 5  Towards Graph Query Autocompletion for Large Graphs  123

5.1 Preliminaries ..................................................... 123
  5.1.1 Background on graph query autocompletion (gQAC) .......... 123
5.2 Wildcard Label for gQAC ......................................... 127
5.3 Query specialization and query summarization .................. 131
  5.3.1 Query specialization ....................................... 132
  5.3.2 Query summarization ...................................... 133
5.4 Autocompletion Framework for Large Graphs - FLAG .......... 134
  5.4.1 Candidate suggestions generation ......................... 135
  5.4.2 Suggestions Ranking ....................................... 137
  5.4.3 Efficient Summary Computation ............................ 140
  5.4.4 Greedy Ranking Algorithm ............................... 141
5.5 Indexed GQAC for Large Graphs - DAGL ....................... 142
  5.5.1 Wildcard Feature DAG (WDAG) Index ...................... 142
  5.5.2 Suggestion Summarization DAG (SSDAG) Index ............. 145
  5.5.3 Autocompletion Using DAGL .............................. 147