Weekly reading report

February 25, 2018

1 Query auto-completion for non-text

There are few literatures specifically on query auto-completion for non-text, [1] is for graph completion for diagram editor. Given a graph H in DIAGEN language, the completion helps to insert new edges into H, or do identification of distinct nodes of H. It uses Cocke algorithm with dynamic programming to finish graph completion and transfer it into diagram completions by computing all possible graph completions up to a user-specified size and offer choices to users to select candidate from editor. [2] is working on subgraph query auto completion (AutoG), the proposed increment for a query are a set of features (c-prime features) where frequency, and construction cost constraints are applied. The algorithm to rank and index the suggestion query are discussed. Their method shows query suggestions in graph saved roughly 40% of users' mouse clicks. The writer here addresses a very similar problem to math-aware search engines, i.e. chemists are not often expected to learn the complex syntax of a graph query language in order to formulate meaningful query over a chemical database such as PubChem or eMolecule. This paper states the auto completion of subgraph queries, to their best knowledge, has not been studied before. Their proposed query auto completion process is: user submit a query as well as preference, AutoG returns ranked suggestions according to the preference. They formalize their auto completion problem as RSQ (ranked subgraph query suggestion problem), and prove RSQ is NP-hard. In order to deal with this hardness, author proposes feature DAG (FDAG) with the ability to identify redundant suggestions via graph automorphism, and their ranking scheme is in favor of high structural diversity. In their paper, they formally defined subgraph isomorphism and subgraph query which is interestingly very similar to the problem definition of math formula subtree search in my thesis. [3] cites grammar-based technique, Spatial Relation Graph based and Spatial Division Tree based method to predict user's intent. This paper uses a set of template symbols forms the symbol dictionary and a spatial relation descriptor on couple primitives with information of the position of one with respect to the other, these features form polar histograms to be used to measure similarity between two graphs, and also consider the subgraph isomorphisms between partially down symbol to template symbol. Their method is tested on a subset of Military Course of Action Diagrams dataset, obtaining above 70% recognition rate. [4] addresses the issue in XML data query where XPath or XQuery language is unfriendly to non-expert users, they introduce auto-complete into XML search and argue type-ahead search can provide instant feedback and thus better experience to users without typing complete keywords. Their system is based on variant (ELCA) of lowest common ancestor of XML tree where LCA is the nearest ancestor of a group of query nodes. They use trie index and edit distance threshold to correct user's incomplete query, so they call this predicted words or fuzzy type-ahead search. What worth mentioning here is they uses the longest prefix measurement to find the cached keyword results, and furthermore, they union merge the posting list corresponding to keywords with same prefixes in fuzzy search. I tried to understand further on their method but I cannot follow their examples since I find many numbers used in their example inconsistent to my understanding.

2 Theory papers on subgraph matching

Because two papers from above section use subgraph matching as metrics of their auto-completion candidate similarity, I did a research on theory papers on tree matching and subtree isomorphism. [5] is a survey paper on subtree matching in particular, it recalls several similar problems, for subtree isomorphism search, the basic algorithm complexity is $O(k^{1.5}n)$ for finding all the subtrees that are isomorphic between a pattern and subject tree where k and n are the number of vertices in the two trees respectively. And the subtree homeomorphism variation where deleting nodes and adding edges are allowed, can be solved by the same algorithm. Another similar problem, tree inclusion, is NP-hard. [6] is focusing on the case of rooted, unordered subtree isomorphism with degrees bounded by a constant d. Their paper lists survey and textbooks for tree-similarity

problem, or largest common subtree problem which has been studied for past a few decades. Their main result is giving a lower bound for this problem depending on Strong Exponential Time Hypothesis (SETH). [7] describes the basic algorithm and also a faster version proposed for finding subtree isomorphism.

Reference

1. S. Mazanek, S. Maier, and M. Minas, "Auto-completion for diagram editors based on graph grammars," in 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, 2008, pp. 242–245.

2. P. Yi, B. Choi, S. S. Bhowmick, and J. Xu, "AutoG: a visual query autocompletion framework for graph databases," The VLDB Journal, vol. 26, no. 3, pp. 347–372, Jun. 2017.

3. M. D. Rosa, "On the auto-completion of hand drawn symbols," in 2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2012, pp. 223–224.

4. Feng, J., and G. Li. 2012. "Efficient Fuzzy Type-Ahead Search in XML Data." IEEE Transactions on Knowledge and Data Engineering 24(5): 882–95.

5. Cserkuti, P., T. Levendovszky, and H. Charaf. 2006. "Survey on Subtree Matching." In 2006 International Conference on Intelligent Engineering Systems, , 216–21.

6. Abboud, Amir et al. 2016. "Subtree Isomorphism Revisited." In SODA '16, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1256–1271. (February 25, 2018).

7. Shamir, Ron, and Dekel Tsur. 1999. "Faster Subtree Isomorphism." Journal of Algorithms 33(2): 267–80. http://www.sciencedirect.com/science/article/pii/S0196677499910441 (February 25, 2018).