Automatic Attribute Ranking in a Keyword Search Result Using Mutual Information

Yodsawalai Chodpathumwan, University of Illinois at Urbana-Champaign
ADVISOR: Arash Termehchy, Prof. Marianne Winslett

Background

- IMDB hires a web-programmer to build a search engine on a movie database. All information attributes of each movie are presented and ranked manually by the programmer.
- A current demo of CRSearch [1] presents a list of search results, and all attributes of each result are presented to the user. This makes page results very long, and the user would not like to scroll down from the top to the bottom of the page just to find the expected answer.
- Based on statistics and probability, we can compute the correlation value between any two attributes of an entity in the dataset, called Mutual Information score (MI score).

Goals

- Based on MI score and user’s keywords query, construct an automatic way to generate a presentation of search results with their attributes in ranked order on an xml database.

Fundamental Questions/Challenges

- How to determine the importance of each entity’s attribute in each candidate search result.
- How to present the results based on ranked attributes.
- The performance should not be penalized because of this new development.

Research Plan

- Examine the mutual information scores for the database
- Implement the algorithm according to the technique employed
- Implement and add the program script to the current system
- Deploy the website demo
- Get users’ feedback

Terminology

- An LCA tree is a database subtree whose leaves contain only query keyword nodes.
- A pattern tree is a subtree that has two leaves representing a pair of any possible attributes of any entity in the database. Each pattern tree has a field that contains an MI score.

Technique Employed

- We develop a technique that creates an LCA-Pattern tree by combining an LCA tree and pattern trees that contain at least one path from root to leaf in the LCA tree based on a modification of the minimum common supertree problem. Also, the number of attributes filled out by the database according to the constructed LCA-Pattern tree at a level is limited by the value of specified topK.

Research Results

- This implementation can create a pattern tree that ranks attributes based on their mutual information.
- The performance of the implementation has a large initial cost, as it needs to compute the LCA-Pattern tree for every answer of a query. However, the cost of fetching the content of the tree from the database is reduced, as it only partially fetches some data from the database.

Future Tasks

- Deploying web demo and getting user feedback
- More optimization on building LCA-Pattern tree.

Related Work