

GExp Manual

July 1, 2018

Contents

1	The Components in the GExp	2
1.1	Configuration Panel	3
1.2	Search Panel	4
1.3	Suggestion Panel	5
2	How to Use	6
2.1	How to Search	6
2.2	Datasets	7
2.3	Scenario 1	8
2.4	Scenario 2	11
3	Deployment on GExp	14
3.1	Keyword Search Classes	14
3.2	GUI Main	14

1 The Components in the GExp

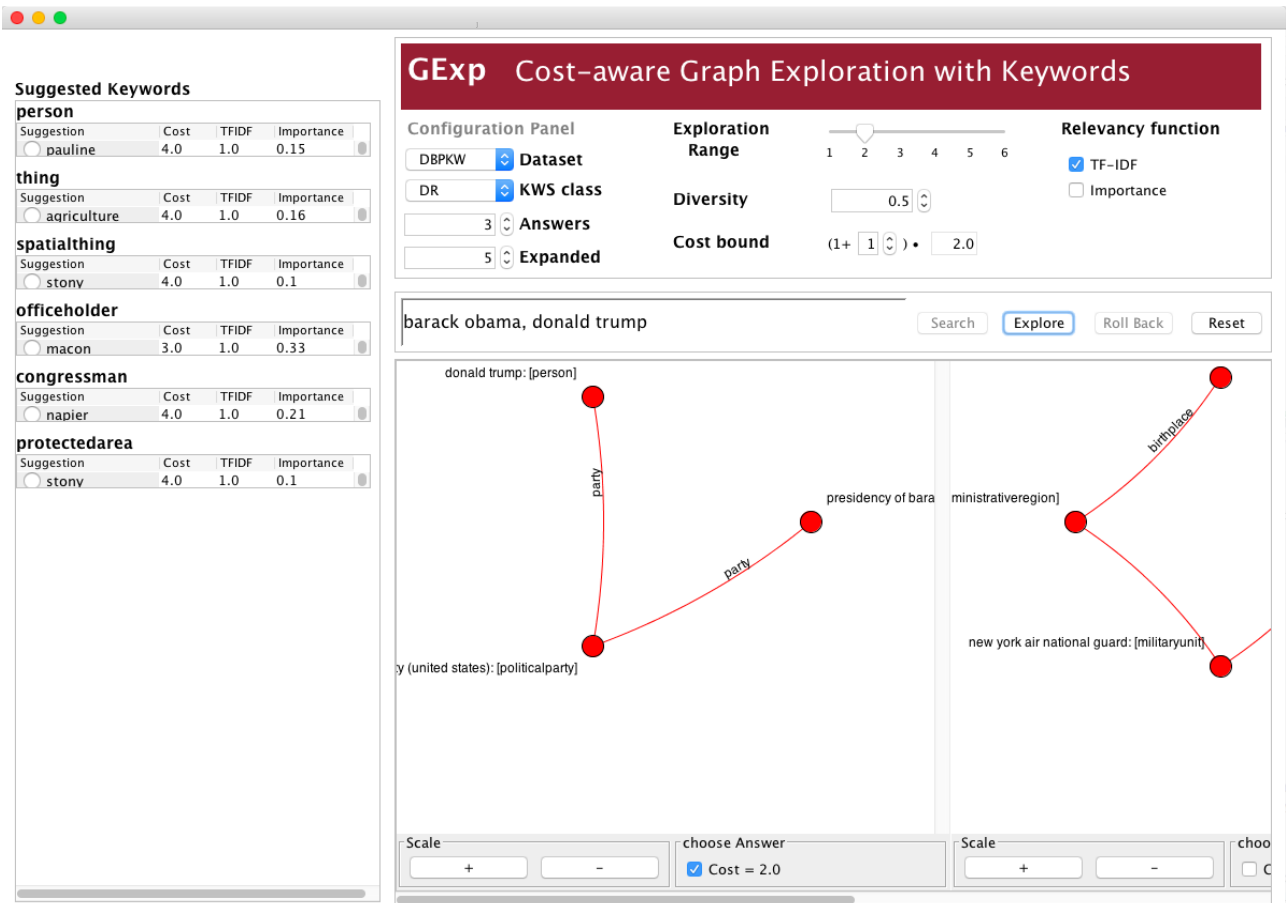
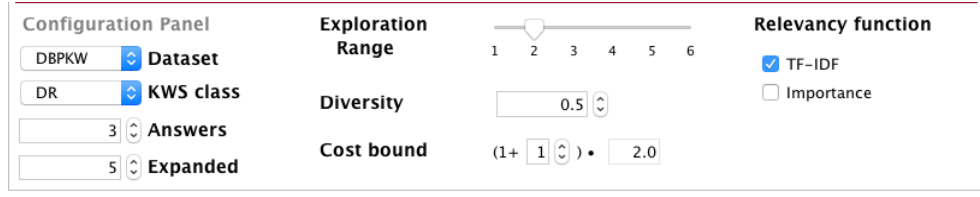


Figure 1: GExp GUI

Figure 1 shows a screenshot of the GExp. The GExp contains three panels that includes configuration, search, and suggestion panels. In this section, we describe the components in each panel.

1.1 Configuration Panel



The Configuration Panel for GExp includes the following components:

- Dataset:** A dropdown menu currently set to 'DBPKW'.
- KWS class:** A dropdown menu currently set to 'DR'.
- Answers:** A numeric input field set to '3'.
- Expanded:** A numeric input field set to '5'.
- Exploration Range:** A horizontal slider ranging from 1 to 6, with the marker positioned at 2.
- Diversity:** A numeric input field set to '0.5'.
- Cost bound:** A formula input field showing '(1+ 1) * 2.0'.
- Relevancy function:** Two checkboxes; 'TF-IDF' is checked, and 'Importance' is unchecked.

Figure 2: Configuration Panel

Figure 2 shows the configuration panel of the GExp. It has following configuration components:

- **Datasets:** the user selects a dataset from the list. For now, we have DBpedia and IMDB. The user can customize it if needed (this is hard-coded for now).
- **KWS query classes:**
 - **Distinct root-based queries (DR)** [3] search for trees with distinct root and bounded depth r (exploration range in GExp). The answer cost of a tree G_Q , determined by a cost function $F(G_Q)$, is computed as the sum of distances from each content node (leaf) to its root v_r .
 - **Steiner tree-based queries (ST)** [2] compute minimum weighted Steiner trees with bounded depth r (exploration range in GExp), where the cost $F(G_Q)$ of a tree G_Q is the sum of its edge weights.
 - **Steiner graph-based queries (SG)** [4] that computes r -cliques, where the cost $F(G_Q)$ of an r -clique G_Q is the total pairwise distance among the content nodes.
- **Answers:** The maximum number of top answers that will be shown in decreasing order of their cost.
- **Expanded:** The maximum number of suggested keywords.
- **Exploration Range:** The search bound of KWS query classes.
- **Diversity:** The weight/impact of the diversity. If the user chooses a relevancy function, she can tune diversity parameter in trade-off with a relevancy parameter (For more info, please read [5]).
- **Cost bound:** The cost bound for suggestions.
- **Relevance function:** The user can choose relevance function to obtain different suggestions. By default, the suggestion will be ranked by cost. If the user chooses relevance function, the suggestions will be ranked by a combination of relevance and diversity score.

1.2 Search Panel

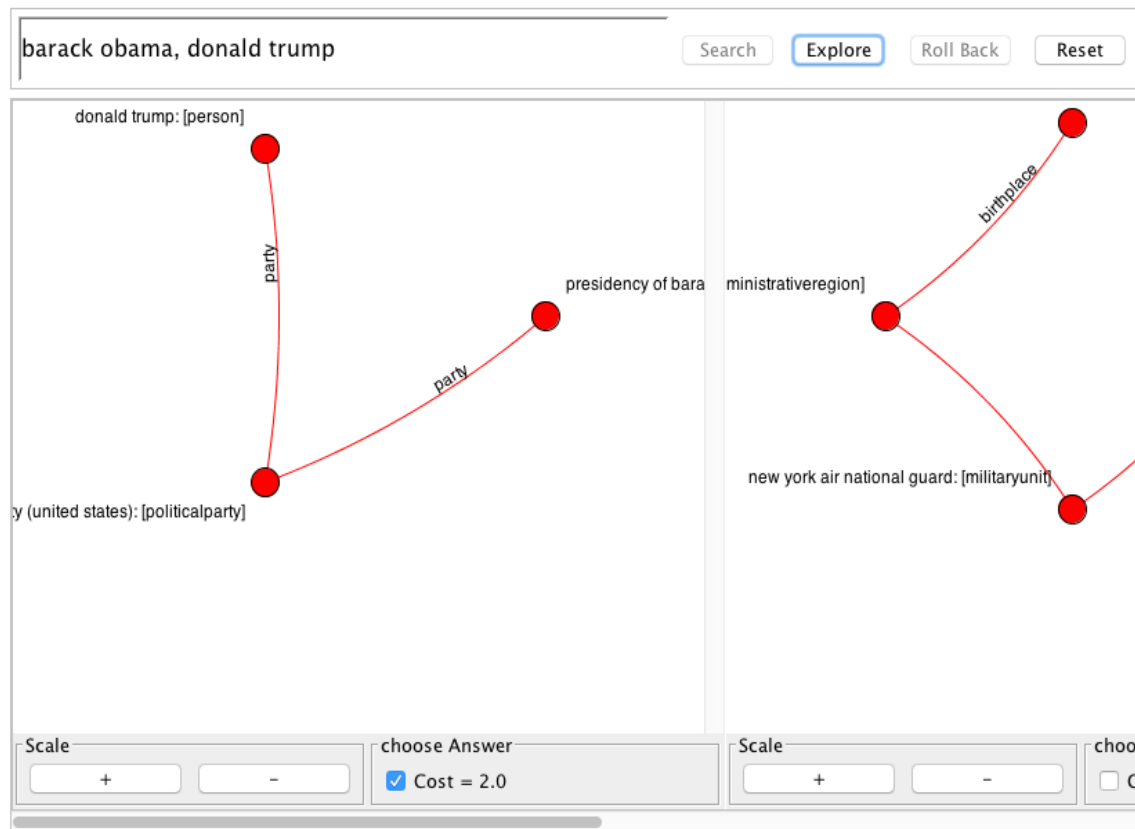


Figure 3: Search Panel

Figure 3 shows the search panel of the GExp.

- **Search box:** A text field for typing query.
- **Search button:** To search for top keyword search results.
- **Explore button:** After selection of a subset of shown answers, a user clicks on this button to obtain suggestions.
- **Roll back button:** After obtaining new results, the user can click roll back button in order to roll back to the previous states.
- **Reset button:** The user can click reset button to start a new round of searching.
- **Result panel:** The results will be shown in this panel. The user can click and drag a node to change the position of each. By clicking on +/– in scale, the user can change the size of each answer. By ticking the checkbox in the choose answer(s) , the user can explore more around the chosen answer(s).

1.3 Suggestion Panel

Suggested Keywords				
person				
Suggestion	Cost	TFIDF	Importance	
<input type="radio"/> pauline	4.0	1.0	0.15	
thing				
Suggestion	Cost	TFIDF	Importance	
<input type="radio"/> agriculture	4.0	1.0	0.16	
spatialthing				
Suggestion	Cost	TFIDF	Importance	
<input type="radio"/> stony	4.0	1.0	0.1	
officeholder				
Suggestion	Cost	TFIDF	Importance	
<input type="radio"/> macon	3.0	1.0	0.33	
congressman				
Suggestion	Cost	TFIDF	Importance	
<input type="radio"/> napier	4.0	1.0	0.21	
protectedarea				
Suggestion	Cost	TFIDF	Importance	
<input type="radio"/> stony	4.0	1.0	0.1	

Figure 4: Suggestion Panel

Figure 4 shows the suggestion panel of the GExp.

- **Suggestion:** After the user clicks explore button, the suggestion panel will show the suggested keywords which are separated in different table by the node type of keywords.
- **Cost:** The new cost after current keyword is added into the chosen answer(s).
- **TFIDF:** The TF-IDF score of current keyword. This score measures the importance of a keyword w.r.t its frequency in the neighbors of answers and its total frequency in data graph G. The higher the frequency is in the neighbors while the lower the frequency is in total G, the higher the TF-IDF is.
- **Importance:** The importance score of current keyword. This score measures the importance of a keyword w.r.t to the degree of nodes contain it. The higher the degree of nodes containing those keyword is, the higher the importance of that keyword is.

2 How to Use

2.1 How to Search

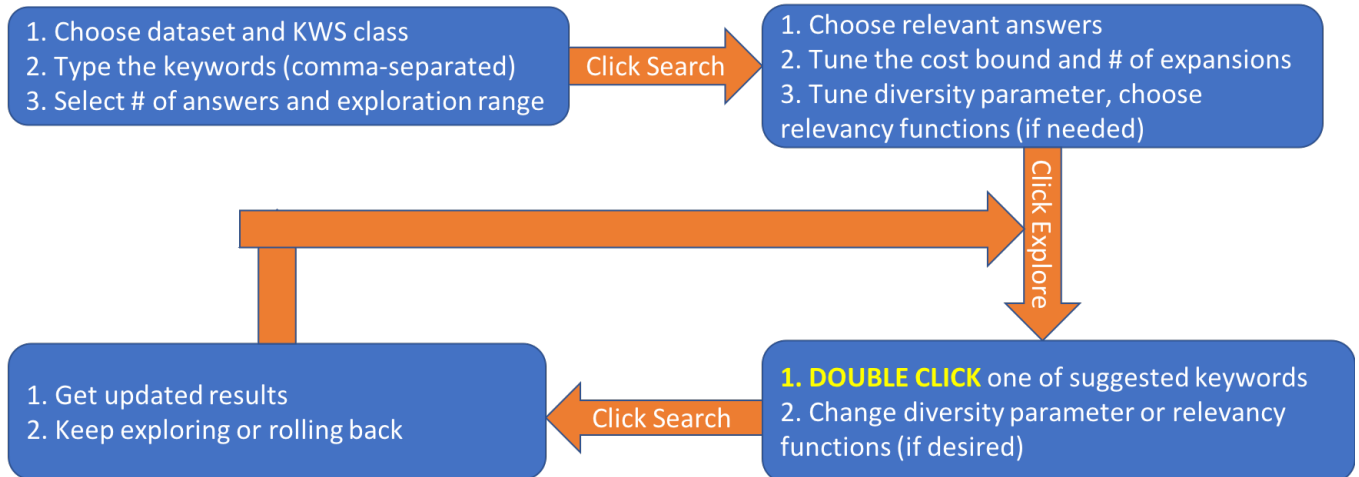


Figure 5: How to Search

Figure 5 shows process of searching in GExp.

1. The user should first choose the dataset and KWS class, then select the answers size, exploration range, and input the keywords. After the user clicks search button, the search panel will show a set of results.
2. If the user wants to explore more, she can choose the interesting results to her and tune the cost bound. Also, the user can choose relevance function and diversity. After the user clicks explore button, the suggestion panel will show suggested keywords.
3. The user can **DOUBLE CLICK** on a suggested keyword and click on the search button. Then, the search panel will show new results. The user can change relevance function or diversity parameter to see different suggestions. After obtaining new results, the user can keep exploring to get expanded answers or rolling back to try other keywords.
4. If the user wants to search other query, she can click reset button.
5. The user should type at least two different keywords and the keywords should be comma-separated. For each keyword that may consisted of multiple tokens like "Jennifer Lopez" that consisted of two tokens, a user may input either "Jennifer", "Lopez", or "Jennifer Lopez". The first one searches for all

nodes contain 'Jennifer' and the last produces less candidates (increasing the search speed/quality).

2.2 Datasets

We provide DBPedia and IMDB as two default datasets. Each entity is a node with type and property information. For example "President" is a type and "Barack Obama" is a value of a property (name). Relationships between entities shown by an edge with type information. For example, "Michelle Obama" is *spouse of* "Barack Obama".

- The DBPedia dataset¹, extracted from Wikipedia, is a knowledge base. Each node in the graph is an entity which is a page in Wikipedia such as a person, country, or a school, etc. Each edge in the graph represents a relation between two entities such as born in, or graduated from, etc.
- The IMDB, collected from IMDB website, is a movie rating dataset. Each node in the graph represents an actor/actress, a movie, or a genre. An edge between two nodes represents relation between an actor and a movie, or between a movie and a genre, etc.

If the user wants to add other datasets, she can put the dataset files in: "../GraphExamples/demo" where "GraphExamples" is under "APEQP" directory (root of the project). The minimal files for a dataset includes "vertices.in" and "relationships.in" that contain nodes and relationships, respectively. To make use of SG, a pruned landmark index [1] should be preprocessed.

The format of the dataset should follow the existing datasets in "demo" directory.

¹<http://dbpedia.org>

2.3 Scenario 1

The user can run scenario 1 to tune the exploration. We also demonstrate how a user can tune the parameters cost bound and exploration range to control the exploration at any time. The user first set up configuration panel as follows:

- Dataset: DBPedia
- KWS class: SG
- Answers: 3
- Exploration Range: 1
- Query: knuth, award, stanford

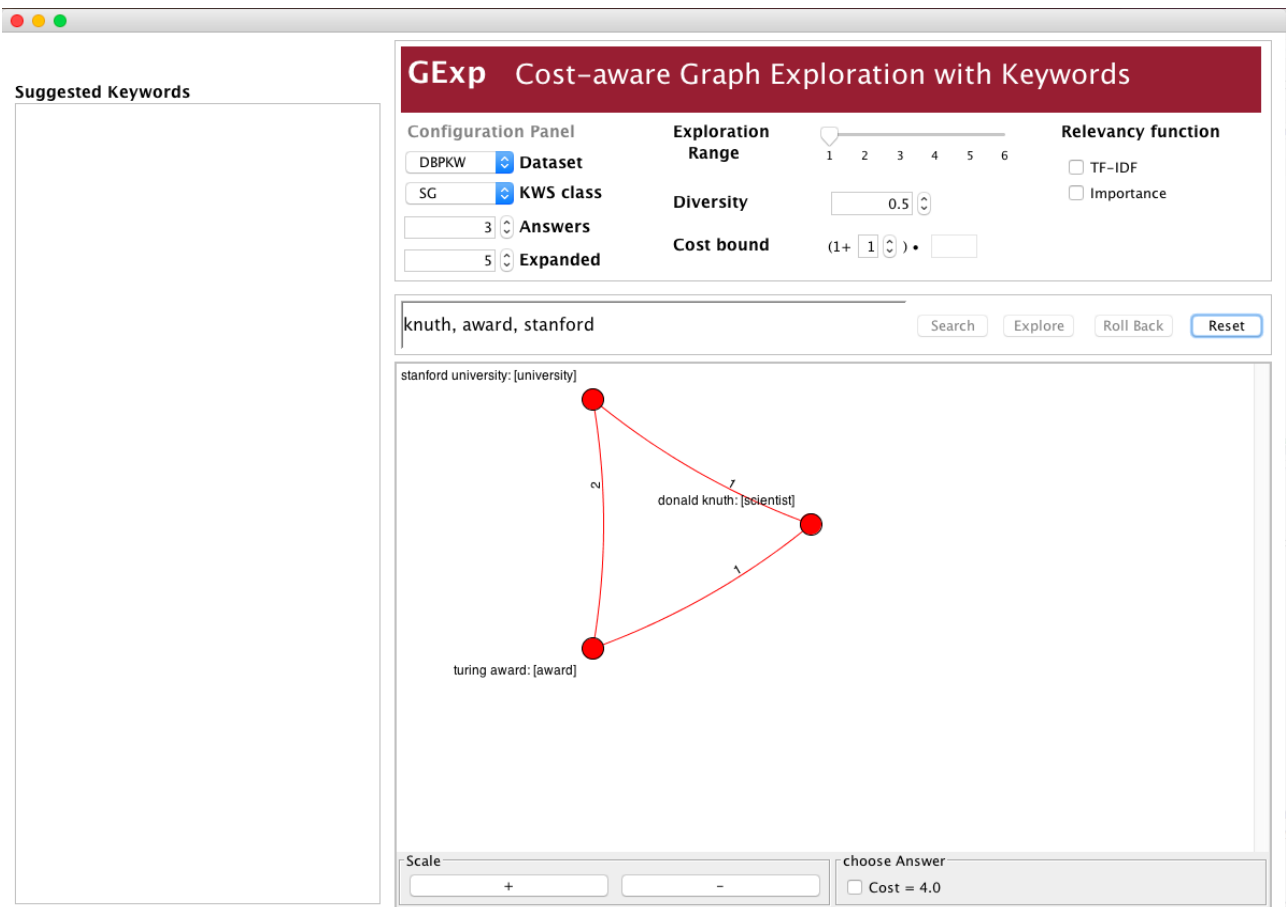


Figure 6: The result of Query: knuth, award, stanford over DBPedia dataset and SG

With exploration range =1 and cost bound =1, the user can find the scientist Sedgewick as suggestion.
The user can set up configuration panel as follows:

- Exploration Range: 1
- Cost bound: 1

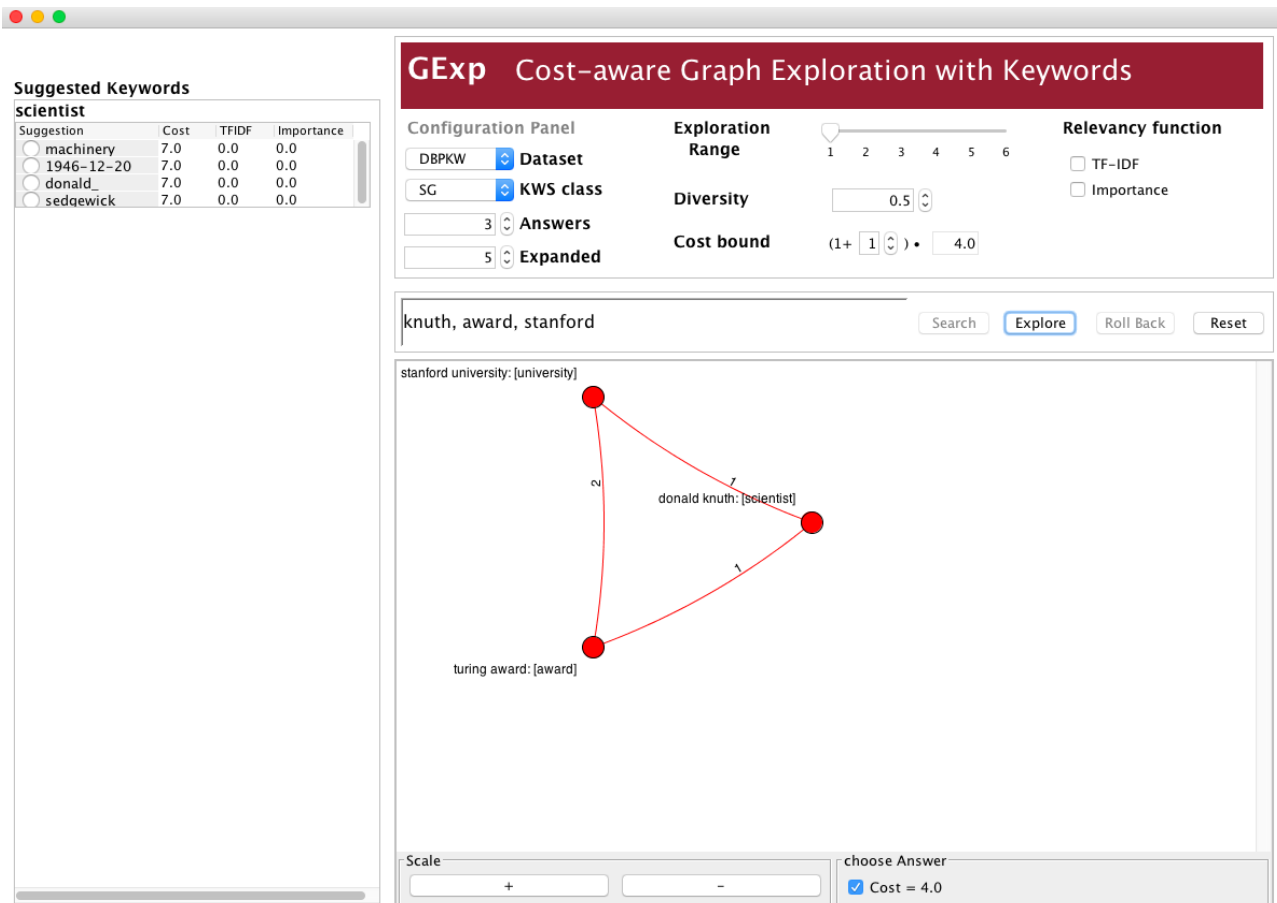


Figure 7: The suggestions over exploration range = 1 and cost bound =1

With exploration range =2 and cost bound =1, the user can find the scientist Ron Rivest as suggestion.
The user can set up configuration panel as follows:

- Exploration Range: 2
- Cost bound: 1

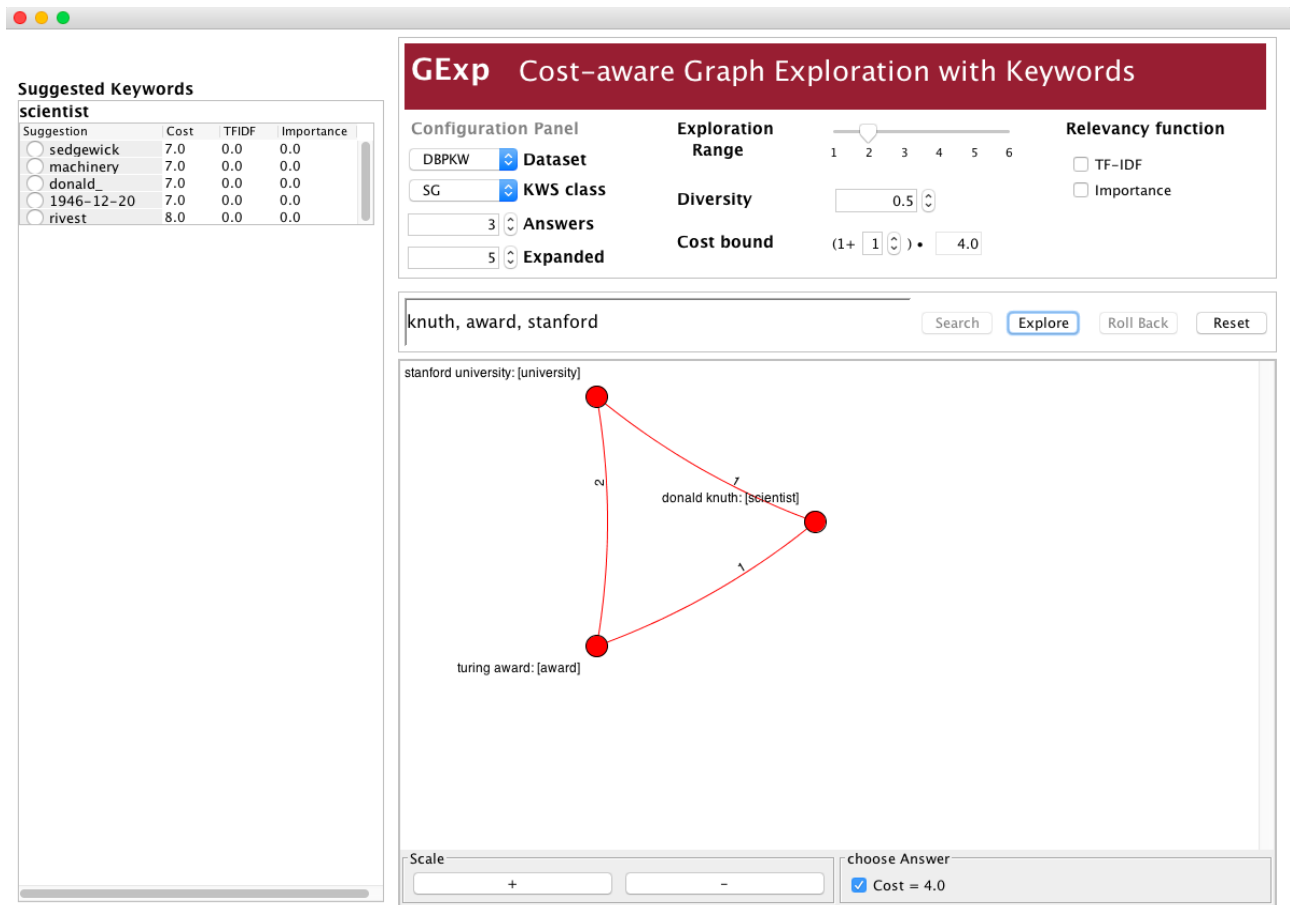


Figure 8: The suggestions over exploration range = 2 and cost bound =1

2.4 Scenario 2

The user first set up configuration panel as follows:

- Dataset: IMDB
- KWS class: DR
- Answers: 2
- Exploration Range: 1
- Query: Jessica Chastain, Anne Hathaway

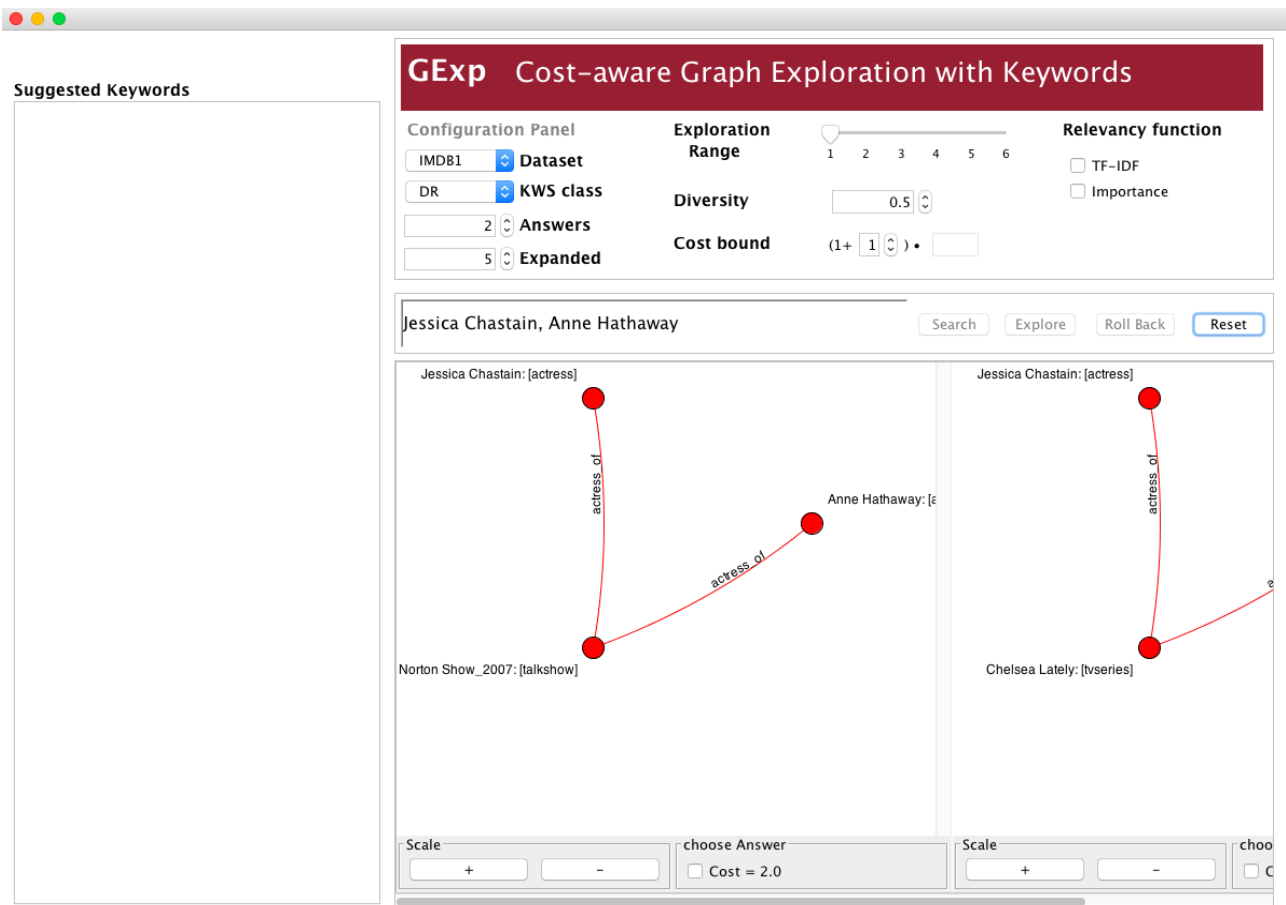


Figure 9: The result of Query: Jessica Chastain, Anne Hathaway over IMDB dataset and DR

With exploration range =1 and TFIDF as relevance function, the user can find the Taylor Swift and Comedy as suggestions.

The user can set up configuration panel as follows:

- Expanded: 16
- Exploration Range: 1
- Relevance function: TFIDF

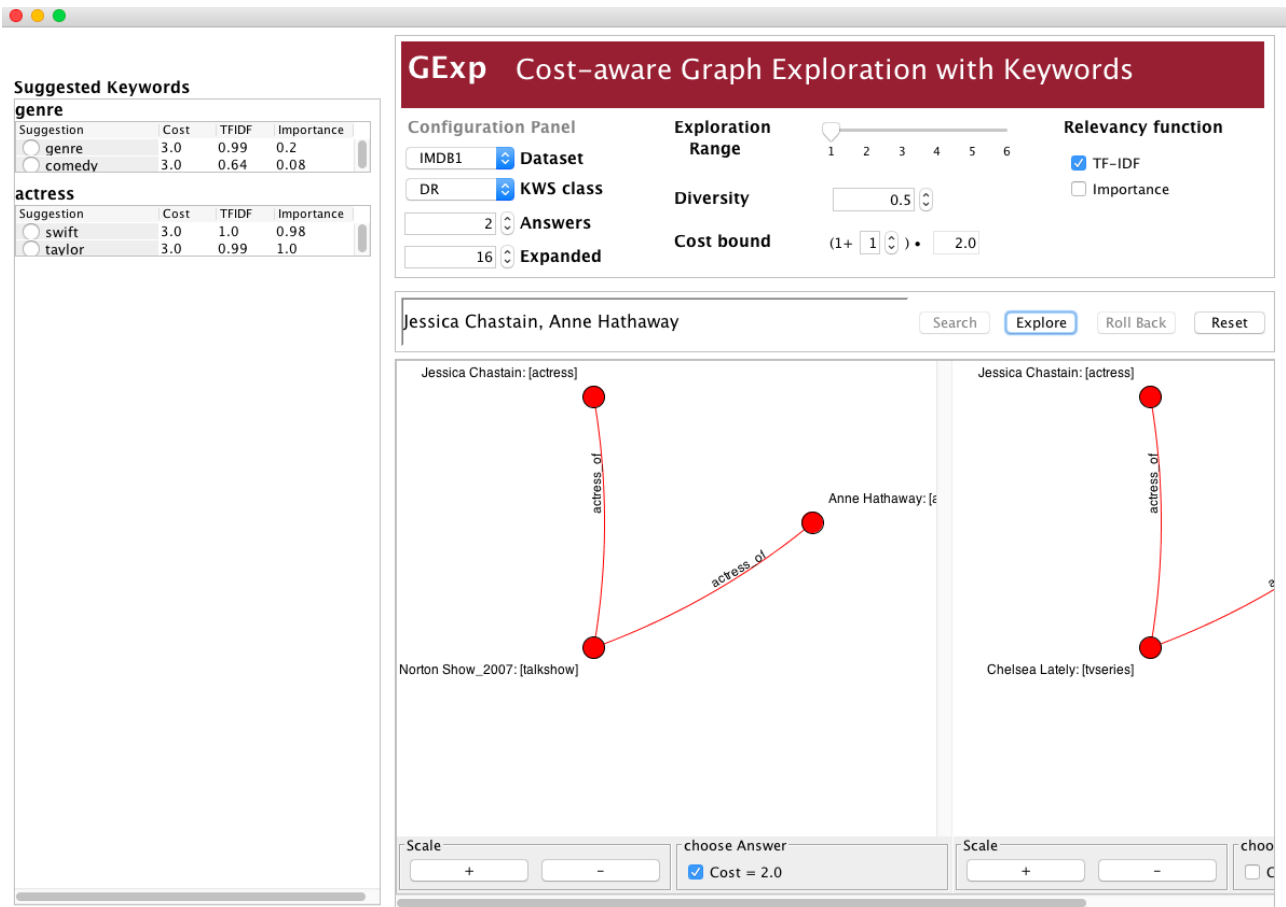


Figure 10: The suggestions over exploration range = 1 and TFIDF as relevance function

With exploration range =2 and TFIDF as relevance function, the user can find the Spanish TV series titled El hormiguero, which provides the user with new information about actresses who co-played in specific movies with those in Q1.

The user can set up configuration panel as follows:

- Expanded: 16
- Exploration Range: 2
- Relevance function: TFIDF

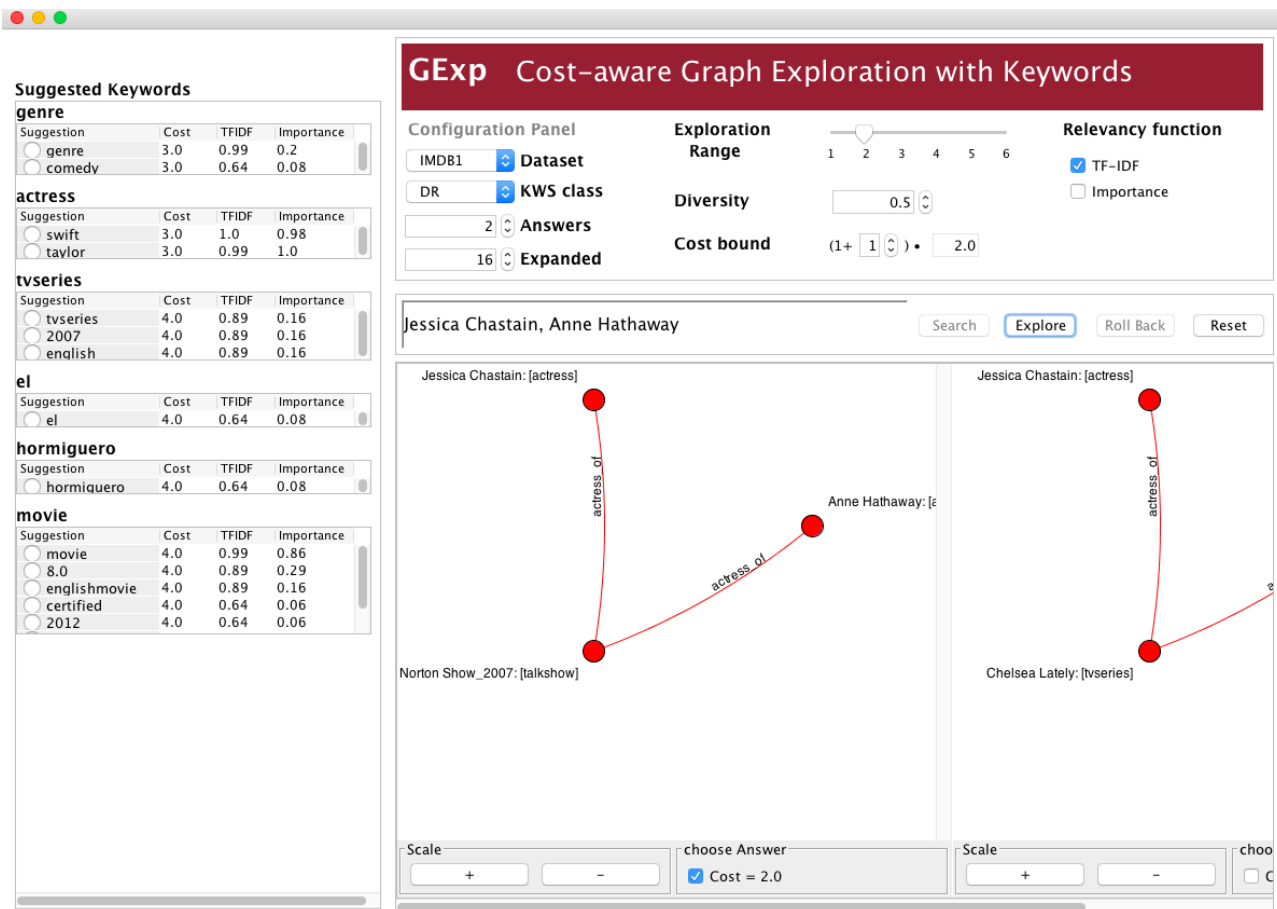


Figure 11: The suggestions over exploration range = 2 and TFIDF as relevance function

3 Deployment on GExp

3.1 Keyword Search Classes

There are three KWS classes: DR, ST, and SG.

- The code for DR: `bank.keywordSearch.DRDemo.java`
- The code for ST: `steiner.keywordSearch.SteinerbasedKWSDemo.java`
- The code for SG: `pairwiseBasedKWS.PairwiseKeywordSearchDemo.java`

3.2 GUI Main

The code for GUI Main: `demo.DemoGUI.java`

The user can run `DemoGUI.java` to use the GExp system.

Reference

- [1] Takuya Akiba, Yoichi Iwata, and Yuichi Yoshida. Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In *SIGMOD*, pages 349–360, 2013.
- [2] Bolin Ding, Jeffrey Xu Yu, Shan Wang, Lu Qin, Xiao Zhang, and Xuemin Lin. Finding top-k min-cost connected trees in databases. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 836–845. IEEE, 2007.
- [3] Varun Kacholia, Shashank Pandit, Soumen Chakrabarti, S Sudarshan, Rushi Desai, and Hrishikesh Karambelkar. Bidirectional expansion for keyword search on graph databases. In *Proceedings of the 31st international conference on Very large data bases*, pages 505–516. VLDB Endowment, 2005.
- [4] Mehdi Kargar and Aijun An. Keyword search in graphs: Finding r-cliques. *Proceedings of the VLDB Endowment*, 4(10):681–692, 2011.
- [5] Mohammad Hossein Namaki, Yinghui Wu, and Xin Zhang. Gexp: Cost-aware graph exploration with keywords. In *SIGMOD*, pages 1729–1732, 2018.