The manual of the visualization tool

GUOKUN LAI
laiguokun@163.com

2015 summer

1 Introduction to the project

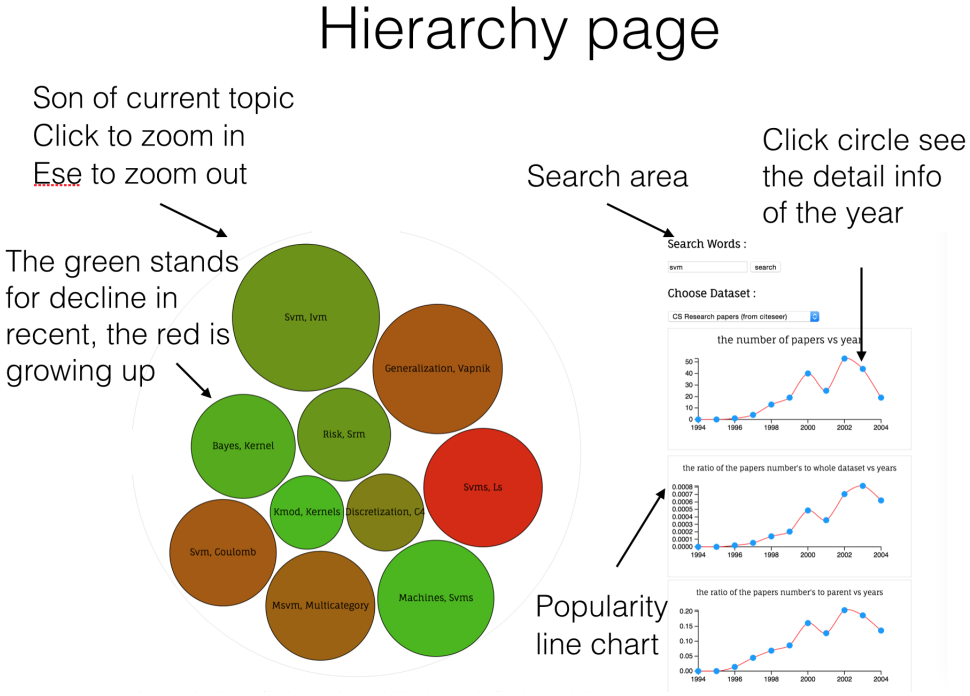
This project is designed to visualize the hierarchical and temporal topic model.

The server is coded by python language. The website is mainly implemented by javascript. The interaction between server and web is depended on ajax. The graph of web is built on d3 library.

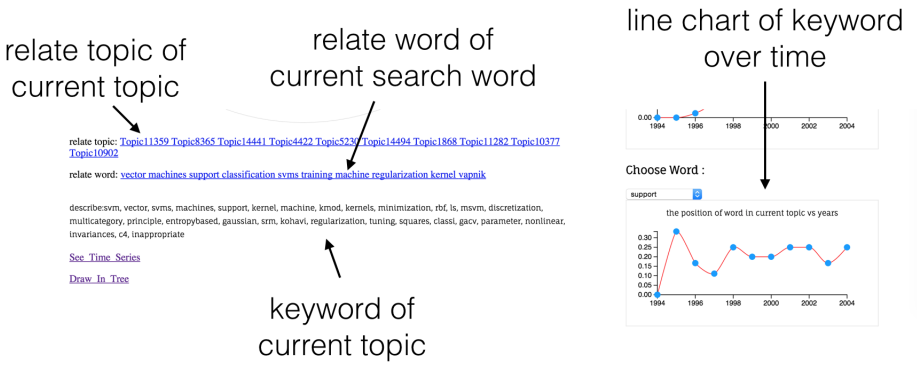
If you have any questions, you can contact me by the email.

2 Usage of the visualization tool

2.1 gc.html

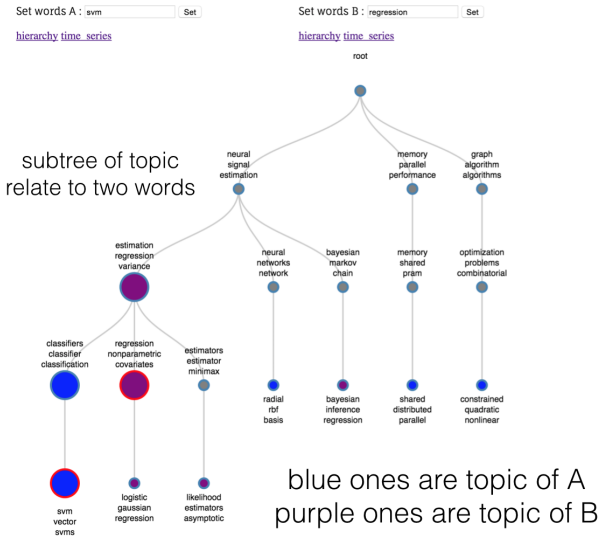


the bottom of hierarchy page



The similarity of topics is defined as the cosine similarity of the word distribution of the topics.
The similarity of words is defined as the times that they occur simultaneously in the keyword lists.

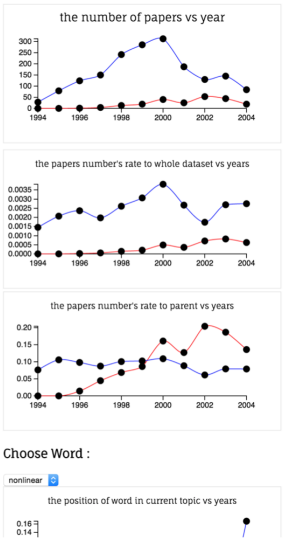
Compare Page



subtree of topic
relate to two words

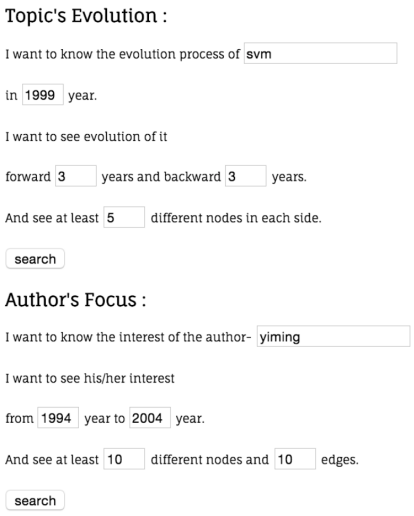
blue ones are topic of A
purple ones are topic of B

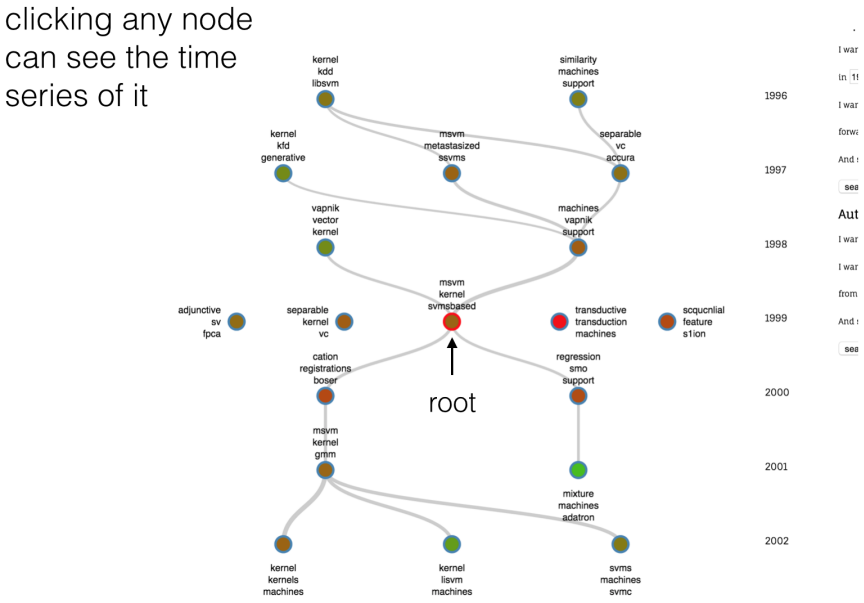
compare the popularity



Time-Series page request

At left, it is the request designed for the user to explore the time series charismatic of the dataset. The first is about topic's evolution over time. The second is to reveal the specific author focus change over time.





2.2 compare.html

3 File of the project

run_server.py is the server of the website.
config.cfg is the config file of project.
gc.html/css/js is the website file of the hierarchy page.
compare.html/css/js is the website file of the compare page.
ts.html/css/js is the websit file of the time series page.

4 Start the project

Run the run_server.py, then you can assess the website from the assigned port.

5 Config

The config file of the project is config.fig. The file contains the information about the direction of the database that use in the project. It also contains the minimum year of the dataset and the maximum year of the dataset. And it has the port number of the project.

6 Database

In this section, I will describe the main content of every dataset used in the server. First at all, I use the leveldb dataset in this project. And in follow, I will give the sample of the index and what info belongs to every kind of index. As for the detail struct of the dataset, you can assess every dataset by python to know that. You can base on these database to build new database of new dataset.

6.1 MetaDB

This database is mainly about the hierarchy struct of the dataset. The index is *csxml_children_nodeid*. It contain the children's id of this node, the keyword lists of children and whether this node is leaf or not.

6.2 ContentDB

This database includes the article of the dataset. The index is *csxml_content_nodeid*. The nodeid in this index comes from the children's id of the leave node in the MetaDB. In this dataset, every item should has the date of the article in $YYYY - MM - DD$ format.

6.3 TimeLineDB

This DB includes the number of papers of every year for every topic. It has three kinds of the index. The index of the first one is *nodeid_sum*, this kind of items have the number of papers of all years for this topic. The second one is *nodeid_r*0, this kind of times have the ratio of the papers' number to the whole dataset. The last one is *nodeid_rp*, this kind of times have the ratio of the papers' number to its parent.

Note that for every time it should have the data for all years for minimum year to maximum year parameter of the config file.

6.4 WordSeriesDB

This DB has two kinds of data. (The two kinds of the data are a little irrelevant.)

The index of the first one is *nodeid_year*. This one has the detail information of the topic in a year. The *desc* attribution is the keyword list of topic. The *keyw* attribution is reranked, considered the idf, keyword list, used as the title of topic.

The index of the second one is *nodeid*. This one describe how the rank of keywords change over year, the keyword of here is the word has occurred in the keyword list of this topic. And I use the inverse of rank in this database.

6.5 DescDB

This DB includes the keyword list for each topic. The index is *nodeid*.

6.6 RelateDB

This DB is used to build the time series graph. The index is *nodeid_year*. For each item, it has two attribution. The first one is *pre*, and it contains the 20 most similar topic of *nodeid* in the year previous to index. The second one is *next*, and it contains the 20 most similar in the year next to index.

6.7 AuthorDB

This DB stores the relate topic for an author. The index is $author's name$. The item contains the set of pair $(nodeid\_year, rank\_of\_professor)$.

6.8 ReferenceDB

This DB stores the relate topic for a reference. The index is $reference's name$. The item also contains the set of pairs $(nodeid\_year, rank\_of\_reference)$

6.9 RelateWordAndTopicDB

This DB includes two kinds of index. The first one is *topic_nodeid*. The terms of this index contains the most similar topic in the dataset for the index. The second one is $word\_(the_word)$. The terms of this index contain the most similar word in the vocabulary for the index.

7 Function of Implement

In this section, we will introduce the main function of the javascript. If you have no interesting in the implement, you can just skip this section.

7.1 gc.js

render_timeLine() is the function to draw the line-chart about the popularity trend of current topic.

render_nonleaf() is the function to draw the hierarchy structure of the dataset.

render_leaf() is the function to show the document for the leaf topic.

7.2 compare.js

render_timeLine() is the function to draw the line-chart about the popularity trend of topic A and topic B.

render_tree() is the function to draw the tree structure of the hierarchy structure.

7.3 ts.js

render_tree() is the function to draw the time series information of the hierarchy structure.