

SHANGHAI JIAO TONG UNIVERSITY

INTRODUCTION TO ELECTRICAL ENGINEERING (B)

Final Project

Author: Wang BoLUN

June 24, 2018



Contents

1	Division of Labor	2
2	Front end	2
2.1	Homepage	2
2.2	Other pages	2
3	Conference Page	5
3.1	Contained Information	5
3.2	Function	8
3.3	Front End Beautification	9
4	Paper search	13
4.1	FULLTEXT search	13
4.2	Paper content	14
5	Paper recommendation	17
5.1	recommend papers which refer to the paper displayed	17
5.2	recommend papers in specific conference	18
5.3	recommend papers in specific range of years	18
6	Data Visualizing	19
6.1	Modules Needed	19
6.1.1	D3.js	19
6.1.2	Logistic Regression Classifier	20
6.2	Characteristic Data Needs Preparing for Prediction	20
6.3	Data Set Preparing	21
6.3.1	Ajax Request Sending	21
6.3.2	Backend Processing	21
6.3.3	Result Display	23
6.3.4	Insert.py	23
6.3.5	Fetch Data for Tree Graph	25
6.4	Graph Drawing	26
6.4.1	Force-directed Graph	26
6.4.2	Tree Graph	28
6.5	Results Display	32
6.5.1	Force-directed Graph	32
6.5.2	Tree Graph	33
7	Achievement Display	33

1 Division of Labor

Front end Home: Wang Kuoye
Conference Page: Wang Bolun
Paper Page: Guo Lingfeng
Author Page: Zheng Longjie

2 Front end

2.1 Homepage

1.Functions: For a search engine, the most important functions we need is a search bar and different buttons which lead to different pages.

To build a homepage which is both beautiful and easy to use. We choose Bootstrap as template.

Bootstrap provides me abundant resources to build a homepage easily.

Since that we have 3 different entries for our websites, which lead to different pages: conferences, papers and authors.

And also, when switching to different modes, the search bar should provide different hints. **2.Solutions:** Bootstrap provides buttons groups , which can satisfy our requirements.

For different buttons, we can use onclick

To finish autocomplete functions according to different pages, I choose to change their names and ids and then use `getElementsby*` to choose them.

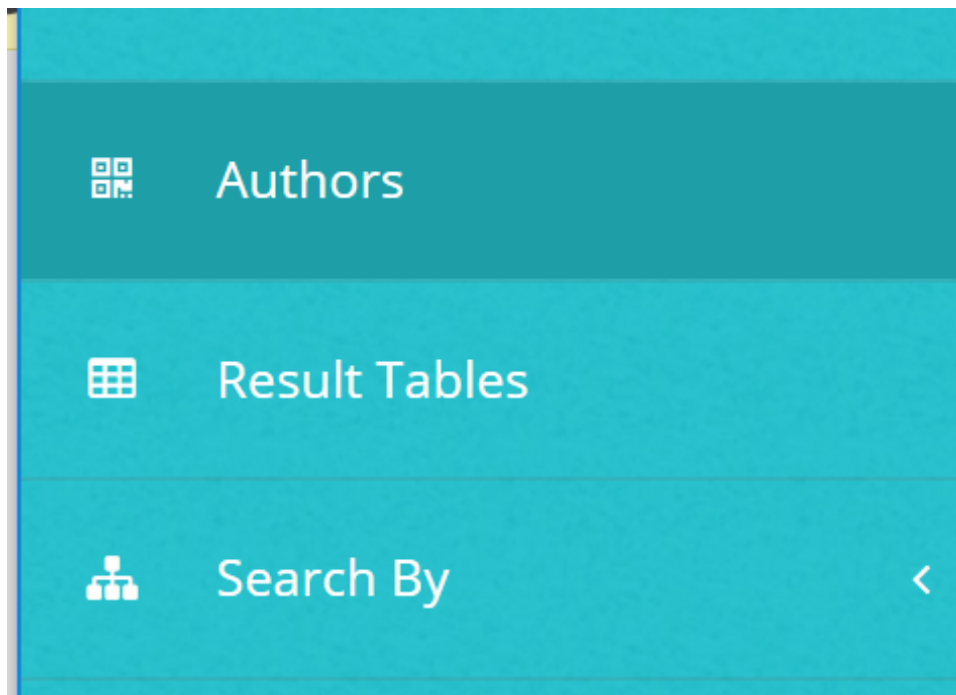
```
1 function turncon()
2 {
3   document.getElementsByName("name")[0].id="id3";
4   document.getElementsByName("name")[0].name="conference";
5   $("#id3").autocomplete({
6     source: "con_buquan.php",
7     minLength: 2,
8     autoFocus:true
9   });
10 }
11 document.getElementsByName("ss")[0].onclick=send3;
```

For more details , you can read `index.php`.

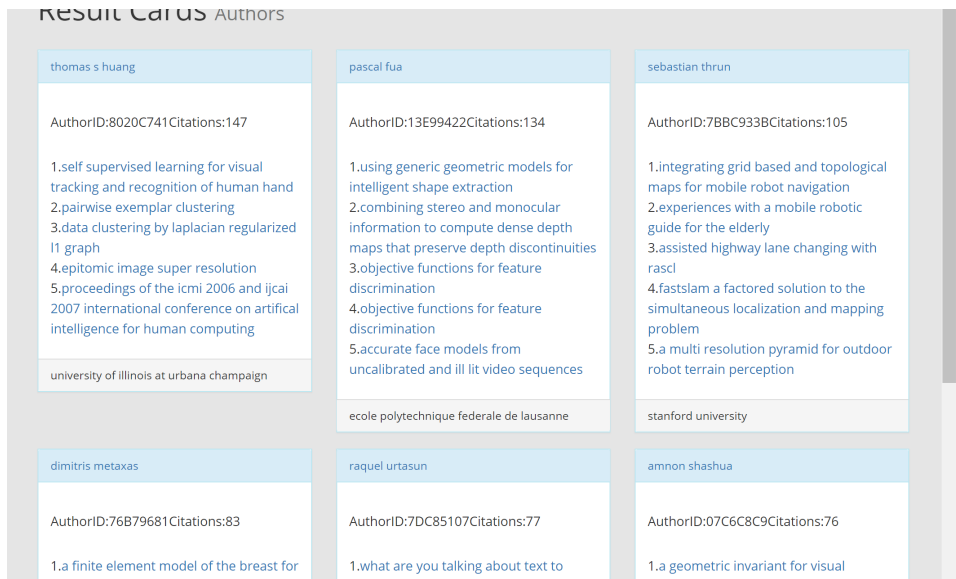
2.2 Other pages

Another part is to beautify different parts:

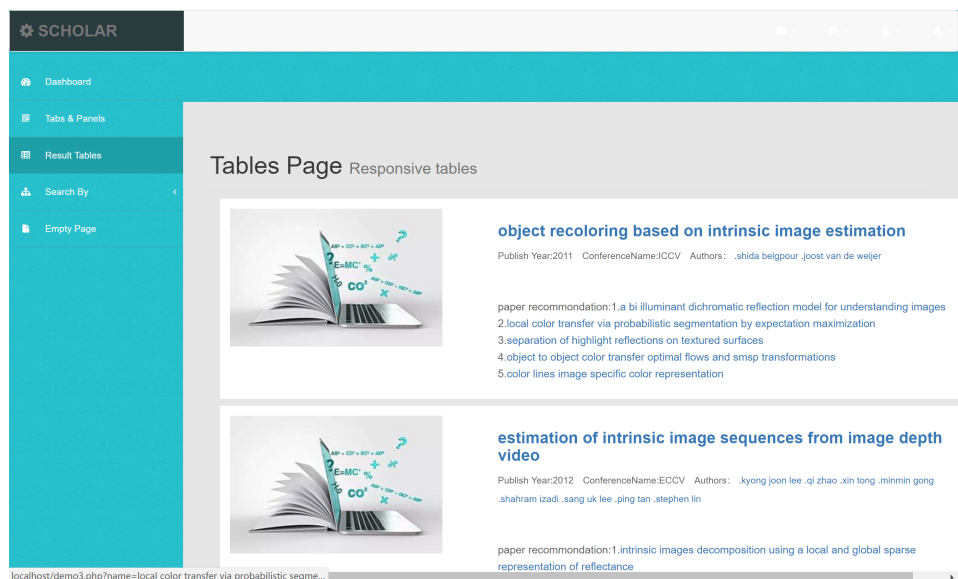
For the whole main page, I design a side bar to navigate in the left:



For the results of authors, I decide to design cards for every author



Its clear to show every authors information one by one.
For the search result,I put them in a table to show them.



This is to unity the theme with my partners.
For search by years I decide to use zebra forms.

Year	Title
2016	a unified energy based framework for learning to rank
2016	transition based dependency parsing with topological fields
2016	visualizing image priors
2016	improved neural network based multi label classification with better initialization leveraging label co occurrence
2016	on logics of strategic ability based on propositional control
2016	privacy preserving class ratio estimation
2016	gated bi directional cnn for object detection
2016	detecting evolution of concepts based on cause effect relationships in online reviews
2016	an mdp based winning approach to autonomous power trading formalization and empirical analysis
2016	map frequency based maximization of airline profits based on an ensemble forecasting approach

I dont want to put a lot of codes here, because front-end has been mixed with back-end. About how contents deliver, you can read tab5.php,table3.php and demo3.php.

3 Conference Page

3.1 Contained Information

1. Brief Introduction:

I download the introduction of every conference from Wikipedia. And use ALTER statements to add a new column `brief_intro` in the `conferences` table, then use UPDATE statements to add the information into the new column. Like this:

```
1 ALTER TABLE conferences ADD brief_intro text
2 UPDATE conferences
3 SET brief_intro=
4 "The International Joint Conference on Artificial Intelligence (IJCAI) is a
   gathering of artificial intelligence researchers and practitioners. It
   is organized by the IJCAI, Inc. It was held biennially in odd-numbered
   years from 1969 to 2015. Starting 2016, IJCAI is held annually. IJCAI
   is a highly selective conference. For instance, only 17 of the papers
   submitted to the conference were accepted in 2011, and in previous
   years never more than 26. This makes it a more selective publication
   than many AI journals.
5 "
6 WHERE ConferenceName='IJCAI'
```

Then when we want to query the brief introduction of a conference, we just need one MySQL statement.

```
1 $conferencef = $_GET["conference"];
2 $brief="select brief_intro from conferences where ConferenceName like '%
   $conferencef%';"
3 $intro=mysql_query($brief);
4 $result0=mysql_fetch_array($intro);
5 echo $result0["brief_intro"];
```

2. Statistical Graph:

In this part, I want to show the number of papers this conference produces every year in a bar graph. It is also convenient to see the general tendency from year to year and display the specific data when the mouse hovers over a column.

Firstly, I need to count the number of every year. That means I should know the conference publish papers in which years and the amount of papers respectively in those years.

```
1 $nianfen="
2 select distinct PaperPublishYear from papers left join conferences
3 on papers.ConferenceID=conferences.ConferenceID where ConferenceName like
   '%$conferencef%'
4 order by PaperPublishYear";
5 $niancha=mysql_query($nianfen);
6
7 while($resultnian=mysql_fetch_array($niancha))
8 {
9     $year=$resultnian['PaperPublishYear'];
10    $years[]=$year;
11    $numyear="select count(*) from papers where PaperPublishYear='{ $year }'";
```

```

12 $numofyear=mysql_query($numyear);
13 $resultnum[]=mysql_fetch_row($numofyear);
14 }
15 foreach($resultnum as $k=>$v)
16 {
17     $datas[]=current($v);
18 }

```

Secondly, I need to draw a chart according to the above data. I use <canvas> tag to take in a chart. And import Chart.min.js from internet to include animated,interactive graphs on my page.

```

1 <script src="skin/Chart.min.js"></script>
2 <canvas id="myChart" width="800" height="300"></canvas>

```

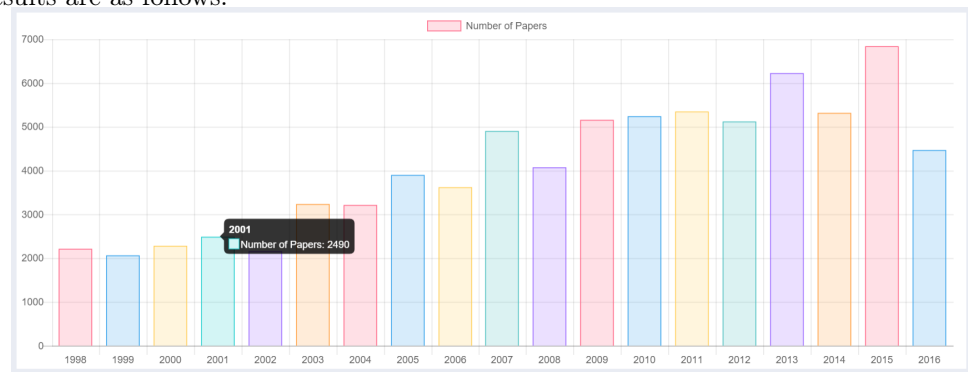
Then use JavaScript to set coordinates.

```

1 <script>
2 var ctx = document.getElementById("myChart").getContext('2d');
3 var myChart = new Chart(ctx, {
4     type: 'bar',
5     data: {
6         labels: eval(<?php echo json_encode($years) ?>),
7         datasets: [{
8             label: 'Number_of_Papers',
9             data: eval(<?php echo json_encode($datas) ?>),
10        }]
11    },
12    options: {
13        scales: {
14            yAxes: [{
15                ticks: {
16                    beginAtZero:true
17                }
18            }]
19        }
20    }
21 });
22 </script>

```

Results are as follows:



3.Paper Information:

I will show some papers' information this conference produces including paper title, publish year, author name and paper picture. That's not difficult and similar to the previous project expect more information.

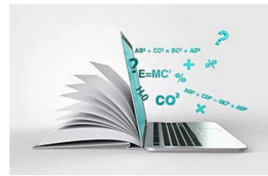
```

1  <?php
2  $sql1="SELECT_ConferenceName,title,PaperPublishYear_FROM_papers_LEFT_JOIN
      conferences_on_papers.ConferenceID=conferences.ConferenceID_where_
      conferences.ConferenceName_like_ '%$conference%'_LIMIT_ $page,10";
3  $result1=mysql_query($sql1);
4  $sql3="SELECT*_FROM_papers_LEFT_JOIN_conferences_on_papers.ConferenceID=
      conferences.ConferenceID_where_conferences.ConferenceName_like_ '%'
      $conference%'_";
5  $result3=mysql_query($sql3);
6  ?>
7
8  <div class="container">
9      <?php while($row1=mysql_fetch_array($result1)){ ?>
10     <?php $title=$row1["title"];?>
11         <div class="list_clearfix_bgw_box">
12             <div class="list_img_pull-left">
13                 <a href="#"></a>
14             </div>
15             <div class="list-box">
16                 <h3><a href="paper.php?title=<?php echo $row1['title']"><?php
                     echo $row1['title'] ?></a></h3>
17                 <div class="info">
18
19                     <em>Publish Year:<?php echo $row1["PaperPublishYear"] ?></em>
20                 <?php
21                 $sql2="select_AuthorName,authors.AuthorID_from_authors_Left_join_
                        paper_author_affiliation_on_authors.AuthorID=
                        paper_author_affiliation.AuthorID_LEFT_join_papers_on_
                        paper_author_affiliation.PaperID=papers.PaperID_where_papers.title_
                        ='$title'";
22                 $result2=mysql_query($sql2);
23                 ?>
24                 <br>
25                 <span>Authors<?php $num=0;while($row2=mysql_fetch_array($result2)){
                        $num=$num+1;?>
26
27                     <a href="author1.php?id=<?php echo $row2['AuthorID']">&page=0"><?php
                        echo "$num_". $row2["AuthorName"] ?></a>
28                 <?php } ?>
29                 </span>
30
31             </div>
32
33         </div>
34     </div>
35 <?php }?>

```

Hint: I leave a position for the pictures and summary about papers. If I was given the detailed information , I can show the abstract and pictures.

Results are as follows:



investigating continuous space language models for machine translation quality estimation

Publish Year:2015

Authors: 1 raymond w m ng 2 kashif shah 3 fethi bougares 4 lucia specia

3.2 Function

1. Automatic Complement:

Similar as project 3 ,only a little different in the inquiry part which the author name was replaced by conference name.

```
1 <?php
2 $sql1="SELECT_ConferenceName_FROM_conferences_WHERE_ConferenceName_like_{
3     $conferencename}%_";
4 $result1=mysql_query($sql1);
5 while ($row=mysql_fetch_array($result1))
6 {
7     $result[] = array(
8         'label' => $row['ConferenceName']
9     );
10 echo json_encode($result);
11 ?>
```

2. Page Turning:

I show 10 pieces of information in each page. And set two buttons to finish the page turning function. The "«" means last page and "»" means next page.

```
1 <ul class="pagination">
2     <li><a href="javascript:previous();">&laquo;</a></li>
3     <li><a href="javascript:next();">&raquo;</a></li>
4 </ul>
```

The Javascript part is same as project 4.

```
1 <script>
2 function next()
3 {
4     page = page + 1 * 10;
5     $.get("http://localhost/conference.php?conference=" + conferenced + "&page=" + page,
6         function(data,status){
7             $("#content").html(data);
8         });
9 }
10 function previous()
11 {
12     if (page <= 0) {
13         return;
14     }
15     page = page - 1 * 10;
```

```

16  $.get("http://localhost/conference.php?conference=" + conferenced + "&page
    =" + page,
17      function(data,status){
18          $("#content").html(data);
19      });
20  }
21  </script>

```

What's more, I add number of total page and current page between these two buttons. In order to know the number of total page, I need traverse all piece of data and count it. But it may cause speed slowly, and it is meaningless and unnecessary that every time we traverse all items. So we can just traverse it once and store it in database. That will be much more quickly.

```

1  $sql3="SELECT*FROMpapers LEFT JOIN conferences on papers.ConferenceID=
    conferences.ConferenceID where conferences.ConferenceName like '%
    $conferenced%'";
2  $result3=mysql_query($sql3);
3  $pages=mysql_num_rows($result3);
4  $a=floor($pages/10);
5  $b=$pages%10;
6
7  <li><a href="#"><?php echo $page/10+1 ?>/
8      <?php if($b==0){
9          echo $a;
10         }else{
11             echo $a+1;
12         } ?></a></li>

```

Effect is as follows:



3. Hyperlink Jump:

When clicking on the author name or paper title, it can skip to the corresponding Author page or Paper page. Besides, when clicking on the conference name in other pages, it can also skip to the corresponding conference page.

3.3 Front End Beautification

I use a template to make the page clean and tidy. Set an overall tone for the web page.

```

1  <link rel="stylesheet" href="skin/bootstrap.min.css">

```

And set some styles in the head including background color, font family, font size, align, margin, padding, border to make the page look comfortably.

```

1 <style>
2 body{background: #e9ecf3;font-family: "HELVETICA NEUE",HELVETICA,ARIAL,SANS-
    SERIF;}
3 .title{color: white;
4     background-color: black;
5     font-size: 35px;
6     align :center;
7
8 }
9 .note-info {
10     background-color: #f5f8fd;
11     color: #010407;
12 }
13 .note {
14     margin: 0 0 20px;
15     padding: 15px 30px 15px 15px;
16     border-left: 5px solid #8bb4e7;
17     border-radius: 0 4px 4px 0;
18     font-size: 13px;
19 }
20 .note h3 {
21     margin-bottom: 10px;
22     font-size: 20px;
23     font-weight: 500;
24 }
25 .note p {
26     margin: 0;
27     font-size: 14px;
28     line-height: 26px;
29 }
30 .bgw{ background-color:#fff; }
31 .box{ padding:15px;border-radius: 0 4px 4px 0;}
32 .list{ margin-bottom:10px;}
33 .list-img{ width:400px;}
34 .list-box{ margin-left:400px;}
35 .list-box{ color:#666; LINE-HEIGHT:200%;}
36 .list-box h3{ font-weight:bold}
37 .info{ margin-bottom:10px;}
38 .info em{ font-style:normal; padding-right:10px;}
39 .info span{}
40 </style>

```

Then use many <div> tags to accomplish every specific region. For example,use 'title' class to show the headline.

```

1 <div class='container'>
2     <h3 class='title' >Conference:<?php echo $conferencef ?></h3>
3 </div>

```

Use 'note note-info' class to show the brief introduction.

```

1 <div class="_container">
2     <div class="note_note-info">
3         <h3>Brief Introduction:</h3>
4         <p><?php echo $result0["brief_intro"] ?></p>
5     </div>
6 </div>

```

Use 'bgw' class to show the graph part. Hint:The border and the inner are same in color, but different in transparency.So it makes people more pleasant to look at.

```

1 <div class="container">
2 <div class="bgw" style="margin-bottom:20px;">
3 <canvas id="myChart" width="800" height="300"></canvas>
4 <script>
5 var ctx = document.getElementById("myChart").getContext('2d');
6 var myChart = new Chart(ctx, {
7   type: 'bar',
8   data: {
9     labels: eval(<?php echo json_encode($years) ?>),
10    datasets: [{
11      label: 'Number_of_Papers',
12      data: eval(<?php echo json_encode($datas) ?>),
13      backgroundColor: [
14        'rgba(255,99,132,0.2)', 'rgba(54,162,235,0.2)',
15        'rgba(255,206,86,0.2)', 'rgba(75,192,192,0.2)',
16        'rgba(153,102,255,0.2)', 'rgba(255,159,64,0.2)',
17        'rgba(255,99,132,0.2)', 'rgba(54,162,235,0.2)',
18        'rgba(255,206,86,0.2)', 'rgba(75,192,192,0.2)',
19        'rgba(153,102,255,0.2)', 'rgba(255,99,132,0.2)',
20        'rgba(54,162,235,0.2)', 'rgba(255,206,86,0.2)',
21        'rgba(75,192,192,0.2)', 'rgba(153,102,255,0.2)',
22        'rgba(255,159,64,0.2)', 'rgba(255,99,132,0.2)',
23        'rgba(54,162,235,0.2)', 'rgba(255,206,86,0.2)',
24        'rgba(75,192,192,0.2)', 'rgba(153,102,255,0.2)',
25        'rgba(255,99,132,0.2)', 'rgba(54,162,235,0.2)',
26        'rgba(255,206,86,0.2)', 'rgba(75,192,192,0.2)',
27        'rgba(153,102,255,0.2)', 'rgba(255,159,64,0.2)',
28        'rgba(255,99,132,0.2)', 'rgba(54,162,235,0.2)',
29        'rgba(255,206,86,0.2)', 'rgba(75,192,192,0.2)',
30        'rgba(153,102,255,0.2)', 'rgba(255,99,132,0.2)',
31        'rgba(54,162,235,0.2)', 'rgba(255,206,86,0.2)',
32        'rgba(75,192,192,0.2)', 'rgba(153,102,255,0.2)',
33        'rgba(255,159,64,0.2)', 'rgba(255,99,132,0.2)',
34        'rgba(54,162,235,0.2)', 'rgba(255,206,86,0.2)',
35        'rgba(75,192,192,0.2)', 'rgba(153,102,255,0.2)'
36      ],
37
38    },
39    borderColor: [
40      'rgba(255,99,132,1)', 'rgba(54,162,235,1)',
41      'rgba(255,206,86,1)', 'rgba(75,192,192,1)',
42      'rgba(153,102,255,1)', 'rgba(255,159,64,1)',
43      'rgba(255,99,132,1)', 'rgba(54,162,235,1)',
44      'rgba(255,206,86,1)', 'rgba(75,192,192,1)',
45      'rgba(153,102,255,1)', 'rgba(255,99,132,1)',
46      'rgba(54,162,235,1)', 'rgba(255,206,86,1)',
47      'rgba(75,192,192,1)', 'rgba(153,102,255,1)',
48      'rgba(255,159,64,1)', 'rgba(255,99,132,1)',
49      'rgba(54,162,235,1)', 'rgba(255,206,86,1)',
50      'rgba(75,192,192,1)', 'rgba(153,102,255,1)',
51      'rgba(255,99,132,1)', 'rgba(54,162,235,1)',
52      'rgba(255,206,86,1)', 'rgba(75,192,192,1)',
53      'rgba(153,102,255,1)', 'rgba(255,159,64,1)'

```

```

54         'rgba(255,99,132,1)', 'rgba(54,162,235,1)',
55         'rgba(255,206,86,1)', 'rgba(75,192,192,1)',
56         'rgba(153,102,255,1)', 'rgba(255,99,132,1)',
57         'rgba(54,162,235,1)', 'rgba(255,206,86,1)',
58         'rgba(75,192,192,1)', 'rgba(153,102,255,1)',
59         'rgba(255,159,64,1)', 'rgba(255,99,132,1)',
60         'rgba(54,162,235,1)', 'rgba(255,206,86,1)',
61         'rgba(75,192,192,1)', 'rgba(153,102,255,1)',
62     ],
63     borderWidth: 1
64 }
65 },
66 options: {
67     scales: {
68         yAxes: [{
69             ticks: {
70                 beginAtZero: true
71             }
72         }]
73     }
74 }
75 });
76 </script>
77 </div>
78 </div>

```

Use 'pagination' class to show the turning page button:

```

1 <div class="container">
2 <div style="background-color:#FFF;padding:8px 12px;text-align:center">
3 <ul class="pagination">
4     <li><a href="javascript:previous();">&laquo;</a></li>
5     <li><a href="#"><?php echo $page/10+1 ?>/
6     <?php if($b==0){
7         echo $a;
8     }else{
9         echo $a+1;
10    } ?></a></li>
11    <li><a href="javascript:next();">&raquo;</a></li>
12 </ul>
13 </div>
14 </div>

```

Finally, I improve the problem of imbedding.

In the past, sometimes I inlay the php code into html and sometimes inlay the html code into php. That makes the layout disorder and the logic unclear. It is difficult to read again and modify.

Now I choose another way. I will write the first part of php including most of inquiry statements on the front. And then write the html codes on the back. When I need to use the data from php part. I will just inlay the php codes into html. Because I have finish most of php on the front, the imbedding section will not very long and sometimes even one statement. Then it looks clearly and distinct.

4 Paper search

4.1 FULLTEXT search

Full-text search refers to the functionality in SQL Server that supports full-text queries against character-based data. These types of queries can include words and phrases as well as multiple forms of a word or phrase. To support full-text queries, full-text indexes must be implemented on the columns referenced in the query. The columns can be configured with character data types (such as char and varchar) or with binary data types (such as varbinary and image). A full-text index is made up of word tokens that are derived from the text being indexed. For example, if the indexed text contains the phrase tables can include indexes, the full-text index would contain four tokens: tables, can, include, and indexes. Because the list of tokens can be easily searched, full-text queries can quickly locate the necessary records.

Here is the specific code which aims to create full-text index in python :

```
sql="""CREATE TABLE papers
(PaperID char(255) NOT NULL,
Title char(255),
PaperPublishYear char(255),
ConferenceID char(255),
FULLTEXT(Title)) ENGINE=MyISAM"""
```

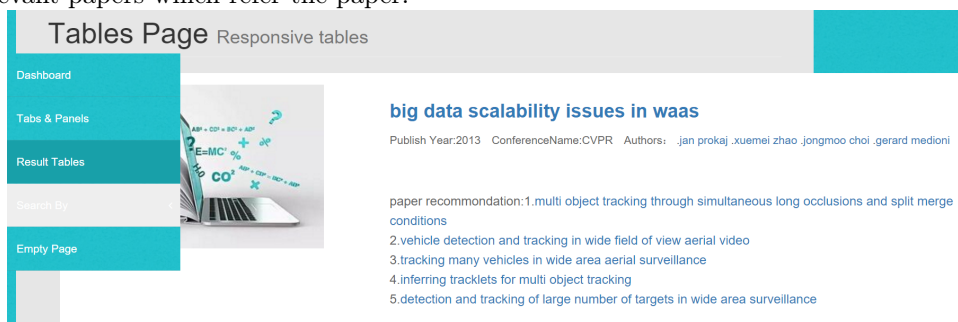
Here is the specific code which uses full-text index to search in html :

```
$sql="SELECT_papers.PaperID , papers.PaperPublishYear ,
ConferenceName , papers.Title
FROM_papers_LEFT_JOIN_conferences
ON_papers.ConferenceID=conferences.ConferenceID
WHERE_Match(Title)_Against_('{$title}');" ;
```



4.2 Paper content

We provide the authors' name, paper's publish year and the conference where paper published according the key words of the paper title and recommend relevant papers which refer the paper.



We also provide navigation bar about the paper's publish year and conferences where paper published to make it more convenient for users to search the relevant papers

Here is the specific codes which create the navigation bar:

```
<li>
<a href="#"><i class="fa fa-sitemap"></i> Search By<span class="fa fa-arrow"></span>
<ul class="nav nav-second-level">
```

```

<li>
<a href="#">Publish Years<span class="fa_arrow"></span></a>
<ul class="nav_nav-third-level">
<li>
<a href=<?php
echo "pd_demo.php?year=2008&title=" .$_GET["name"] . " ' "; ?>>0~10 years </a>
</li>
<li>
<a href=<?php
echo "pd_demo.php?year=1998&title=" .$_GET["name"] . " ' "; ?>>10~20 years </a>
</li>
<li>
<a href=<?php
echo "pd_demo.php?year=1988&title=" .$_GET["name"] . " ' "; ?>>20~30 years </a>
</li>
</ul>
</li>
<li>
<a href="#">Conference<span class="fa_arrow"></span></a>
<ul class="nav_nav-third-level">
<li>
<a href=<?php
echo "pc_demo.php?conference=ECCV&title=" .$_GET["name"] . " ' "; ?>>ECCV</a>
</li>
<li>
<a href=<?php
echo "pc_demo.php?conference=NIPS&title=" .$_GET["name"] . " ' "; ?>>NIPS</a>
</li>
<li>
<a href=<?php
echo "pc_demo.php?conference=SIGKDD&title=" .$_GET["name"] . " ' "; ?>>SIGKDD</a>
</li>
<li>
<a href=<?php
echo "pc_demo.php?conference=WWW&title=" .$_GET["name"] . " ' "; ?>>WWW</a>
</li>
<li>
<a href=<?php
echo "pc_demo.php?conference=SIGIR&title=" .$_GET["name"] . " ' "; ?>>SIGIR</a>
</li>
<li>
<a href=<?php
echo "pc_demo.php?conference=CVPR&title=" .$_GET["name"] . " ' "; ?>>CVPR</a>
</li>
<li>
<a href=<?php

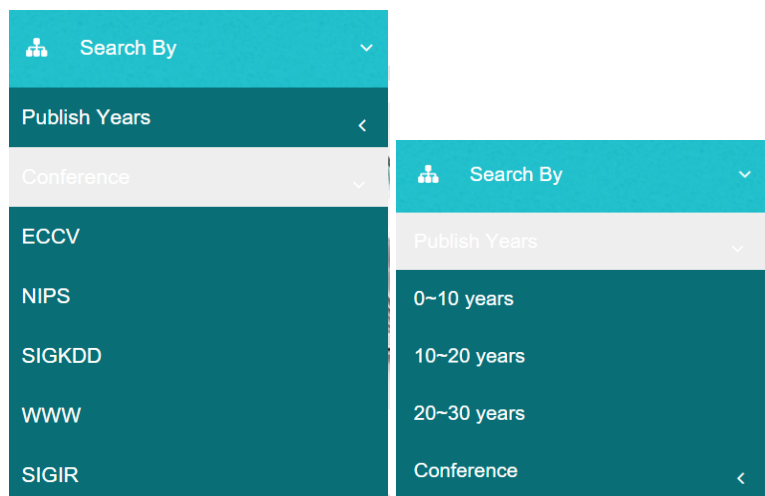
```



```

echo" 'pc_demo.php?conference=ICCV&title=" .$_GET["name"] . " ' "; ?>>ICCV</a>
</li>
<li>
<a href=?php
echo" 'pc_demo.php?conference=NAACL&title=" .$_GET["name"] . " ' "; ?>>NAACL</a>
</li>
<li>
<a href=?php
echo" 'pc_demo.php?conference=ICML&title=" .$_GET["name"] . " ' "; ?>>ICML</a>
</li>
<li>
<a href=?php
echo" 'pc_demo.php?conference=AAAI&title=" .$_GET["name"] . " ' "; ?>>AAAI</a>
</li>
<li>
<a href=?php
echo" 'pc_demo.php?conference=ACL&title=" .$_GET["name"] . " ' "; ?>>ACL</a>
</li>
<li>
<a href=?php
echo" 'pc_demo.php?conference=EMNLP&title=" .$_GET["name"] . " ' "; ?>>EMNLP</a>
</li>
<li>
<a href=?php
echo" 'pc_demo.php?conference=IJCA&title=" .$_GET["name"] . " ' "; ?>>IJCA</a>
</li>
</ul>
</li>
</ul>
</li>

```



We can also click the papers recommended ,then we will turn to new paper page which is searched by the title of paper you clicked.In the same time ,we also add links to the author's name,we can turn to the author page via clicking the author's name.

2016	basic probabilistic ontological data exchange with existential rules
2016	sand semi supervised adaptive novel class detection and classification over data stream
2016	mobility sequence extraction and labeling using sparse cell phone data
2016	big data mechanisms and energy policy design
2016	group and graph joint sparsity for linked data classification
2016	optimizing multivariate performance measures from multi view data

5 Paper recommendation

5.1 recommend papers which refer to the paper displayed

We recommend papers which the papers which refer to the paper displayed and following is the specific code in html:

```
$sql4="SELECT_a. Title
FROM_papers_a_INNER_JOIN
(SELECT_ReferenceID
FROM_paper_reference
WHERE_PaperID_='$paperid')_b
ON_a.PaperID=b.ReferenceID
limit_5
";
```

5.2 recommend papers in specific conference

We recommend papers which is published in the conference you assigned with the relevant paper titles

Here is the specific codes :

```
$sql1="
SELECT title , PaperPublishYear FROM papers
LEFT JOIN conferences
on papers . ConferenceID=conferences . ConferenceID
where conferences . ConferenceName_like_ '$conference%'_and
Match( Title )_Against( _'% $title %' )
ORDER_BY_ PaperPublishYear_DESC
LIMIT_ $page ,10 " ;
```

Year	Title
2016	basic probabilistic ontological data exchange with existential rules
2016	sand semi supervised adaptive novel class detection and classification over data stream
2016	mobility sequence extraction and labeling using sparse cell phone data
2016	big data mechanisms and energy policy design
2016	group and graph joint sparsity for linked data classification
2016	optimizing multivariate performance measures from multi view data

5.3 recommend papers in specific range of years

We recommend papers which is published in the recent ten years, twenty years or thirty years with the relevant paper titles

Here is the specific codes :

```
$sql1="
SELECT title , PaperPublishYear FROM papers LEFT JOIN conferences
on papers . ConferenceID=conferences . ConferenceID
where Match( Title )_Against( _'% $title %' )_and
( PaperPublishYear_between_ $year_and_ ( $year +9))
ORDER_BY_ PaperPublishYear_DESC
LIMIT_ $page ,10 " ;
```

Year	Title
1997	negotiation on data allocation in multi agent environments
1997	some issues in the automatic classification of us patents
1997	learning bayesian networks from incomplete data
1997	summarizing time varying data
1997	presenting and analyzing the results of ai experiments data averaging and data snooping

6 Data Visualizing

Abstract

This part displays the function of data visualizing, which contains the relationship force-directed graph of each author and the tree graph of the predicted teacher-student relationship.

6.1 Modules Needed

d3.js is needed for the force-directed graph and the tree graph, javascript is mainly used to program and python is used to capture and predict the relationship between scholars.

6.1.1 D3.js

Exposition D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3 emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation. D3 allows you to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document. For example, you can use D3 to generate an HTML table from an array of numbers. Or, use the same data to create an interactive SVG bar chart with smooth transitions and interaction.

D3 is not a monolithic framework that seeks to provide every conceivable feature. Instead, D3 solves the crux of the problem: efficient manipulation of documents based on data. This avoids proprietary representation and affords extraordinary flexibility, exposing the full capabilities of web standards such as HTML, SVG, and CSS. With minimal overhead, D3 is extremely fast, supporting large datasets and dynamic behaviors for interaction and animation. D3 functional style allows code reuse through a diverse collection of official and community-developed modules.

Node and link attributes The data set should be performed in json form and the elements in the dataset could be classified as the node or the link. Each node symbolizes an independent element and each link stands for the relationship between two elements, thus providing visible connection between elements. And we could add different attributes to each node or line to personalize each element and relationship.

Citation of D3

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```

6.1.2 Logistic Regression Classifier

Exposition In statistics, the logistic model (or logit model) is a statistical model that is usually taken to apply to a binary dependent variable. In regression analysis, logistic regression or logit regression is estimating the parameters of a logistic model. More formally, a logistic model is one where the log-odds of the probability of an event is a linear combination of independent or predictor variables. The two possible dependent variable values are often labelled as "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick. The binary logistic regression model can be generalized to more than two levels of the dependent variable: categorical outputs with more than two values are modelled by multinomial logistic regression, and if the multiple categories are ordered, by ordinal logistic regression, for example the proportional odds ordinal logistic model.

Logistic regression was developed by statistician David Cox in 1958. The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor. The model itself simply models probability of output in terms of input, and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other. The coefficients are generally not computed by a closed-form expression, unlike linear least squares.

About the Tree Graph By comparing the accuracy of different machine-learning models, i choose logistic model for the highest accuracy rate. And i use python to visit the database to capture the characteristics of each scholar and use the predict function to predict the possible teacher-student relationship and insert the data into the relationship table created in the database.

6.2 Characteristic Data Needs Preparing for Prediction

I capture 9 characteristic values to form a single array and use two-value labels, i use two-dimension numpy arrays to restore all the characteristics and by applying the data set to the predict function, which returns the single-dimension array which restores the predicted labels.

character1:the number of papers published by scholar b before the first cooperation
 character2:the number of papers published by scholar a before the first cooperation
 character3:(character2-character1)/the number of the total cooperated papers
 character4:the gap between the publishyear of the first paper written by a and the
 publishyear of the first cooperated paper
 character5:the gap between the publishyear of the first paper written by b and the
 publishyear of the first cooperated paper
 character6:(character4-character5)/the number of the years cooperated
 character7:the number of papers that a published without cooperating with b during
 the cooperated period
 character8:the number of papers that a published without cooperating with b during
 the cooperated period
 character9:(character7-character8)/the number of the total cooperated papers

6.3 Data Set Preparing

6.3.1 Ajax Request Sending

First we should prepare the data set needed for graph drawing.

```
$.ajax({
    url: "datadeal.php",
    data: {page: page, id: id},
    type: "POST",
    dataType: "JSON",
    success: function(data) {
        console.log(data);
    }
})
```

6.3.2 Backend Processing

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die("Could not connect: " . mysql_error());
}
mysql_select_db("main_db",$con);
mysql_query("set names utf8",$con);
$page = $_POST["page"];
$key = $_POST["id"];
$num = 10;
$tiao = ($page-1)*$num;
$sql = "SELECT paper_author_affiliation.paperid FROM
        (paper_author_affiliation left join paper_reference on
        paper_author_affiliation.paperid=paper_reference.referenceid) left
        join affiliations on
```

```

        paper_author_affiliation.affiliationid=affiliations.affiliationid
        where paper_author_affiliation.authorid=$key GROUP BY
        paper_author_affiliation.paperid ORDER BY COUNT(*) DESC limit
        $tiao,$num";
$query=mysql_query($sql);
$arr=array();
$arr1=array();
$i=0;
while ($row=mysql_fetch_array($query)){
    $id=$row[paperid];
    $result1= mysql_query("SELECT authors.authorid FROM authors left join
        paper_author_affiliation on
        authors.authorid=paper_author_affiliation.authorid where
        paper_author_affiliation.paperid=$id ORDER BY
        paper_author_affiliation.authorsequence ASC");
    while($row1=mysql_fetch_array($result1)){
        $j=0;
        while($j<$i){
            if($arr[$j][id]==$row1[authorid])break;
            $j++;}
        echo $i." ".$j."<br />";
        if($row1[authorid]!=$key && $j==$i){
            $arr[]=array(id=>$row1[authorid],
                group=>1);
            $arr1[]=$row1[authorid];
            $i++;}
        }}
    $arr[]=array(id=>$key,group=>0);
    $arr1[]=$key;
    $n=count($arr1);
    $finalresult=array();
    for($i=0;$i<$n;$i++){
        for($j=$i+1;$j<$n;$j++){
            $newquery=mysql_query("SELECT paper_author_affiliation.paperid
                FROM (paper_author_affiliation left join paper_reference on
                paper_author_affiliation.paperid=paper_reference.referenceid)left
                join affiliations on
                paper_author_affiliation.affiliationid=affiliations.affiliationid
                where paper_author_affiliation.authorid=$arr1[i] GROUP BY
                paper_author_affiliation.paperid ORDER BY COUNT(*) DESC");
            $k=0;
            while($row2=mysql_fetch_array($newquery)){
                $newid=$row2[paperid];
                $newresult=mysql_query("SELECT authors.authorid FROM authors
                    left join paper_author_affiliation on
                    authors.authorid=paper_author_affiliation.authorid
                    where paper_author_affiliation.paperid=$newid ORDER BY
                    paper_author_affiliation.authorsequence ASC");
                while($newrow=mysql_fetch_array($newresult)){
                    if($newrow[authorid]==$arr1[$j]){ $k++;break;}}
            }
        }
    }
}

```

```

    }
    if($k!=0){$finalresult[]=array(
source=>$i,
target=>$j,
value=>$k)}}}
$terminal=array(nodes=>$arr,edges=>$finalresult);
mysql_close($con);
echo json_encode($terminal);
?>

```

6.3.3 Result Display

```

{"id":"7E8C6A66","name":"yun fu","group":1},
{"id":"7FD78E52","name":"shuicheng yan","group":1},
{"id":"85DA3D2C","name":"ming liu","group":1},
{"id":"066D0AF7","name":"mark hasegawajohnson","group":1},
{"id":"7EA72152","name":"na cui","group":1},{"id":"841A6C01","name":"zhen
li","group":1},{"id":"80F98BF8","name":"feng liang","group":1},
{"id":"0D1C47F7","name":"nemanja petrovic","group":1},
{"id":"7CEE9BF0","name":"ira s cohen","group":1},
{"id":"753F7AE6","name":"brendan j frey","group":1},
{"id":"78D26A3C","name":"ralf koetter","group":1},
{"id":"8020C741","group":0,"edges":[{"source":0,"target":1,"value":1},
{"source":0,"target":2,"value":1},{"source":0,"target":16,"value":1},
{"source":0,"target":17,"value":1},{"source":0,"target":18,"value":1},
{"source":0,"target":19,"value":1},{"source":0,"target":20,"value":1},
{"source":0,"target":21,"value":1},{"source":0,"target":26,"value":3},
{"source":1,"target":2,"value":3},{"source":1,"target":11,"value":1},
{"source":1,"target":26,"value":1},{"source":2,"target":11,"value":1},
{"source":2,"target":20,"value":1},{"source":2,"target":26,"value":2},
{"source":3,"target":4,"value":1},{"source":3,"target":5,"value":1},
{"source":3,"target":6,"value":1},{"source":3,"target":26,"value":1},
{"source":4,"target":5,"value":6},{"source":4,"target":6,"value":1},
{"source":4,"target":9,"value":1},{"source":4,"target":26,"value":1},
{"source":5,"target":6,"value":1},{"source":5,"target":9,"value":4},
{"source":5,"target":26,"value":1},{"source":6,"target":26,"value":1},
{"source":7,"target":8,"value":4},{"source":7,"target":26,"value":4},

```

6.3.4 Insert.py

I use the model I trained to scan and predict the whole database and if there is a relationship existing between two scholars,i just insert the data into the relationship table(label 1 for positive and 0 for negative)

```

from sklearn.linear_model import LogisticRegression
from sklearn.externals import joblib
lr=joblib.load("lr.model")
import sys
import pymysql
db =
    pymysql.connect(host="localhost",user="root",db="main_db",password="",charset="utf8",port=3306)
cursor = db.cursor()
def datafetch(id1,id2):
    li=[]
    li1=[]
    li2=[]
    sql1="select paper_author_affiliation.paperid,papers.paperpublishyear
        from paper_author_affiliation left join papers on

```



```

        paper_author_affiliation.paperid=papers.paperid
    where paper_author_affiliation.authorid="\{" order by
        paperpublishyear ASC ".format(id1)
sql2="select paper_author_affiliation.paperid,papers.paperpublishyear
    from paper_author_affiliation left join papers on
        paper_author_affiliation.paperid=papers.paperid
    where paper_author_affiliation.authorid="\{" order by
        paperpublishyear ASC ".format(id2)
try:
    cursor.execute(sql1)
    results1=cursor.fetchall()
    results1=list(results1)
    cursor.execute(sql2)
    results2=cursor.fetchall()
    results2=list(results2)
    for m in range(len(results1)):
        results1[m]=list(results1[m])
        results1[m][1]=int(results1[m][1])
    for n in range(len(results2)):
        results2[n]=list(results2[n])
        results2[n][1]=int(results2[n][1])
    i=0
    while i <=len(results1)-1:
        paperid=results1[i][0]
        j=0
        while j<=len(results2)-1:
            if results2[j][0]==paperid:
                li2.append([results2[j],j])
                break
            else:
                j+=1
        if j<=len(results2)-1:
            li1.append([results1[i],i])
            i+=1
    if len(li1)>0:
        a=(li1[0][1]-li2[0][1])/len(li1)
        b=-results1[0][1]+li1[0][0][1]
        c=-results2[0][1]+li2[0][0][1]
        d=(b-c)/(-li1[0][0][1]+li1[len(li1)-1][0][1]+1)
        e=-li1[0][1]+li1[len(li1)-1][1]+1-len(li1)
        f=-li2[0][1]+li2[len(li2)-1][1]+1-len(li2)
        g=(e-f)/len(li1)
        li.append(li1[0][1])
        li.append(li2[0][1])
        li.append(a)
        li.append(b)
        li.append(c)
        li.append(d)
        li.append(e)
        li.append(f)

```

```

        li.append(g)
    return li
else:
    return 0
except:
    return 0

```

6.3.5 Fetch Data for Tree Graph

```

<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die(Could not connect: . mysql_error());
}
mysql_select_db("main_db",$con);
$id = $_POST["id"];
$name=$_POST["name"];
$arrs=array();
$arrs[0][id]=$id;
$arrs[0][name]=$name;
$arrs[0][children]=array();
$arrt=array();
$arrt[0][id]=$id;
$arrt[0][name]=$name;
$arrt[0][children]=array();
$sqls="select authorid2,authorname2 from relation where
        relation.authorid1=$id and relation.isteacher=1";
$results=mysql_query($sqls);
while($row=mysql_fetch_array($results))
{
    $arrs[0][children][]=array(
        id=>$row[authorid2],
        name=>$row[authorname2]);
}
$sqlt = "select authorid1,authorname1 from relation where
        relation.authorid2=$id and relation.isteacher=1";
$resultt=mysql_query($sqlt);
while($row1=mysql_fetch_array($resultt))
{
    $arrt[0][children][]=array(
        id=>$row1[authorid1],
        name=>$row1[authorname1]);
}
$final=array(teachers=>$arrt,students=>$arrs);
echo json_encode($final);

```

?>

6.4 Graph Drawing

6.4.1 Force-directed Graph

Draw the graph using the data set.

```
var nodes=data.nodes;
var edges=data.edges;
var width = 960;
var height = 600;
var svg= d3.select("center")
    .append("svg")
    .attr("width",width)
    .attr("height",height);

var force = d3.layout.force()
    .nodes(nodes)
    .links(edges)
    .size([width,height])
    .linkDistance(150)
    .charge(-400);
force.start();
var svg_edges = svg.selectAll("line")
    .data(edges)
    .enter()
    .append("line")
    .style("stroke", "#ccc")
    .style("stroke-width",function(d){return
        Math.sqrt(d.value);});

var color = d3.scale.category20();
var svg_nodes = svg.selectAll("circle")
    .data(nodes)
    .enter()
    .append("circle")
    .attr("r",function(d){
        if(d.group==0)return 20;return 10;})
    .style("fill",function(d){
        return color(d.group);
    })
    .on("mouseover",function(d,i){
        edges_text.style("fill-opacity",function(edge){
            if( edge.source === d || edge.target === d ){
                return 1.0;
            }
        });
    })
    .on("mouseout",function(d,i){
        edges_text.style("fill-opacity",function(edge){
            if( edge.source === d || edge.target === d ){
```

```

        return 0.0;
    }
    });
    })
    .call(force.drag);
var svg_texts = svg.selectAll(".nodetext")
    .data(nodes)
    .enter()
    .append("text")
    .attr("class","nodetext")
    .attr("dx", 10)
    .attr("dy", 4)
    .text(function(d){
        return d.id;
    });
var edges_text = svg.selectAll(".linetext")
    .data(edges)
    .enter()
    .append("text")
    .attr("class","linetext")
    .text(function(d){
        return d.value;
    });
force.on("tick", function(){

    svg_edges.attr("x1",function(d){ return d.source.x; })
        .attr("y1",function(d){ return d.source.y; })
        .attr("x2",function(d){ return d.target.x; })
        .attr("y2",function(d){ return d.target.y; });
    edges_text.attr("x",function(d){ return (d.source.x +
        d.target.x) / 2 ; });
    edges_text.attr("y",function(d){ return (d.source.y +
        d.target.y) / 2 ; });

    svg_nodes.attr("cx",function(d){ return d.x; })
        .attr("cy",function(d){ return d.y; });

    svg_texts.attr("x", function(d){ return d.x; })
        .attr("y", function(d){ return d.y; });
    });
},
error:
function(data){
    console.log(data);}
});

```

6.4.2 Tree Graph

```
var treedata1=data.teachers;
var treedata2=data.students;
function treeload(treedata){
var margin = {top: 20, right: 120, bottom: 20, left: 120},
width = 960 - margin.right - margin.left,
height =1200 - margin.top - margin.bottom;

var i = 0,
duration = 750,
root;

var tree = d3.layout.tree()
.size([height, width]);

var diagonal = d3.svg.diagonal()
.projection(function(d) { return [d.y, d.x]; });

var svg = d3.select("center").append("svg")
.attr("width", width + margin.right + margin.left)
.attr("height", height + margin.top + margin.bottom)
.append("g")
.attr("transform", "translate(" + margin.left + "," + margin.top
+ ")");

root = treedata[0];
root.x0 = height / 2;
root.y0 = 0;

update(root);

d3.select(self.frameElement).style("height", "500px");

function update(source) {

var nodes = tree.nodes(root).reverse(),
links = tree.links(nodes);

nodes.forEach(function(d) { d.y = d.depth * 180; });

var node = svg.selectAll("g.node")
.data(nodes, function(d) { return d.id || (d.id = ++i); });

var nodeEnter = node.enter().append("g")
.attr("class", "node")
.attr("transform", function(d) { return "translate(" + source.y0 +
"," + source.x0 + ")"; });
```

```

.on("click", click);

nodeEnter.append("circle")
  .attr("r", 1e-6)
  .style("fill", function(d) { return d._children ? "lightsteelblue"
    : "#357CAE"; });
nodeEnter.append("rect")
  .attr("x", -23)
  .attr("y", -10)
  .attr("width", 70)
  .attr("height", 22)
  .attr("rx", 10)
  .style("fill", "#357CAE");
nodeEnter.append("text")
  .attr("x", function(d) { return d.children || d._children ? 13 :
    13; })
  .attr("dy", "-4")
  .attr("text-anchor", "middle")
  .text(function(d) { return d.id; })
  .style("fill", "white")
  .style("fill-opacity", 1e-6);

nodeEnter.append("line")
  .attr("x1", "-25")
  .attr("y1", "0")
  .attr("x2", "50")
  .attr("y", "0")
  .attr("stroke", "white")

nodeEnter.append("text")
  .attr("x", function(d) { return d.children || d._children ? 13 : 13;
    })
  .attr("dy", "10")
  .attr("text-anchor", "middle")
  .text(function(d) { return d.name; })
  .style("fill", "white")
  .style("fill-opacity", 1);

var nodeUpdate = node.transition()
  .duration(duration)
  .attr("transform", function(d) { return "translate(" + d.y + "," +
    d.x + ")"; });

nodeUpdate.select("rect")
  .attr("x", -23)
  .attr("y", -10)
  .attr("width", 70)
  .attr("height", 22)
  .attr("rx", 10)
  .style("fill", "#357CAE");

```

```

nodeUpdate.select("text")
.attr("text-anchor", "middle")
.style("fill-opacity", 1);

var nodeExit = node.exit().transition()
.duration(duration)
.attr("transform", function(d) { return "translate(" + source.y +
    "," + source.x + ")"; })
.remove();

nodeExit.select("circle")
.attr("r", 1e-6);
nodeExit.select("rect")
.attr("x", -23)
.attr("y", -10)
.attr("width", 70)
.attr("height", 22)
.attr("rx", 10)
.style("fill", "#357CAE");

nodeExit.select("text")
.attr("text-anchor", "middle")
.style("fill-opacity", 1e-6);

var link = svg.selectAll("path.link")
.data(links, function(d) { return d.target.id; });

link.enter().insert("path", "g")
.attr("class", "link")
.attr("d", function(d) {
    var o = {x: source.x0, y: source.y0};
    return diagonal({source: o, target: o});
})
.attr(marker-end, url(#arrow));

link.transition()
.duration(duration)
.attr("d", diagonal);

link.exit().transition()
.duration(duration)
.attr("d", function(d) {
    var o = {x: source.x, y: source.y};
    return diagonal({source: o, target: o});
})
.remove();

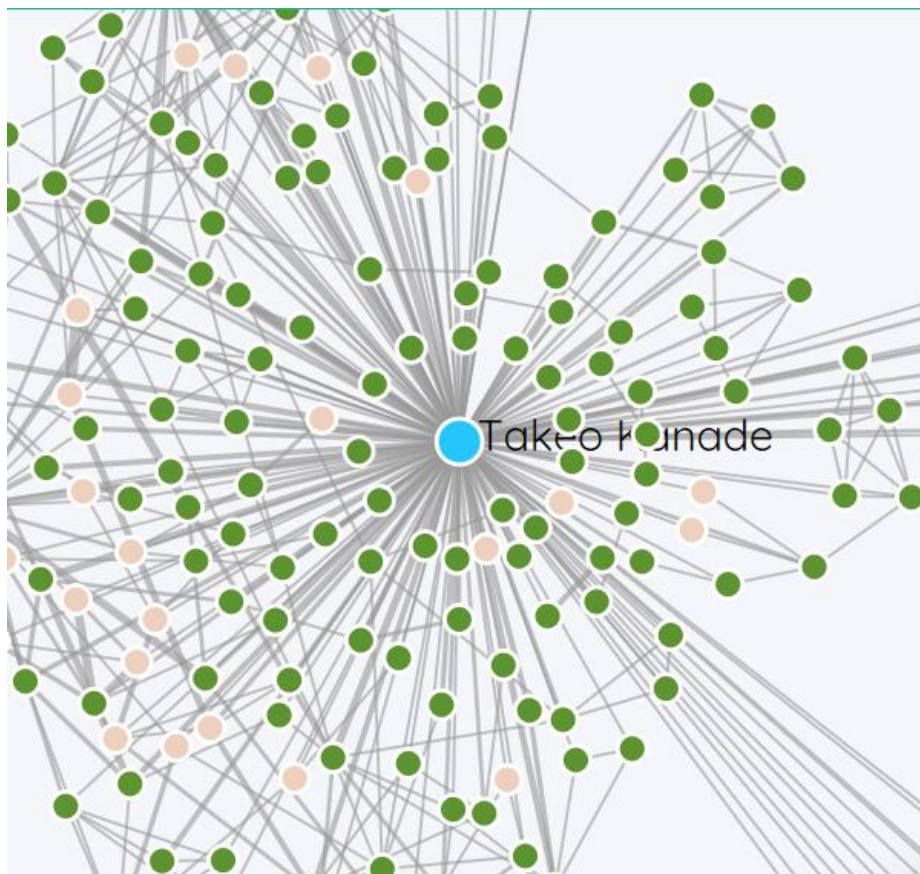
nodes.forEach(function(d) {
d.x0 = d.x;

```

```
        d.y0 = d.y;
    });
}
function click(d) {
    if (d.children) {
        d._children = d.children;
        d.children = null;
    } else {
        d.children = d._children;
        d._children = null;
    }
    update(d);
    }}
    treeload(treedata2);
    treeload(treedata1);
    },
    error:function(data){
        alert("error");});
}
```

6.5 Results Display

6.5.1 Force-directed Graph



6.5.2 Tree Graph

