Experimental Report for Final Project

Group 6

Version: 1

Authors: Xingzi Yu,

Weidong Wang,

QiuXuan ling,

Zhibang Wang

Finished Date: 6 / 14 / 2018

Contact us: Xingzi Yu (edittriendl@sjtu.edu.cn); Weidong Wang (wedonsjtu@sjtu.edu.cn); Qiuxuan Ling (2017lqx@sjtu.cn); Zhibang Wang (wswswzb@hotmail.com)

Contents

Homepage/Result Display/UI Design (Xingzi Yu)

Chapter 1

Homepage

1.1 Description

1.1.1 Brief Introduction to IEEE Search

IEEE Search
is a website focusing on acedamic search. It contains the following pages. (including font-end and back-end pages)

Figure 1.1: Logo of IEEE Search

Figure 1.2: Logo of IEEE Search

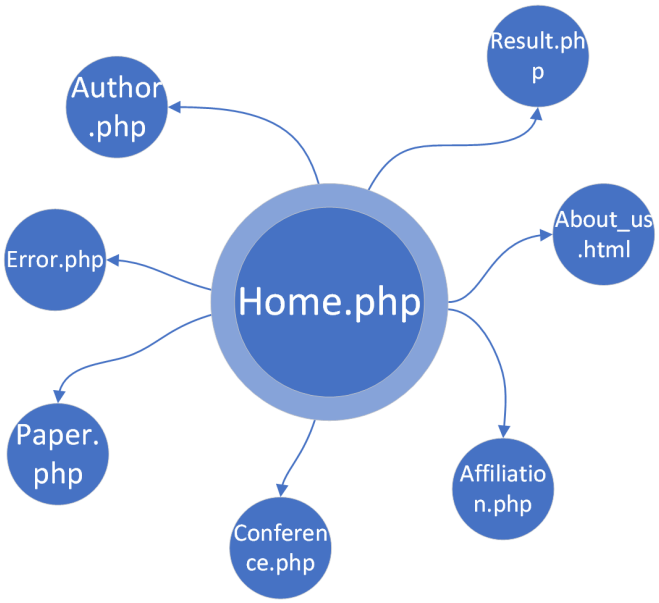


Figure 1.3: Basic Structure of Search Website

As you can see, my team mainly developed 7 pages in IEEE Search Website.

Result.php display the search result of user's input on homepage.

Author.php An author page contains all related information about an particular author.

Paper.php A page displays one paper's information.

Affiliation.php A page of affiliation information.

Conference.php A page of conferences information.

How to Use

In our website, you can search whatever, wherever and whenever you want. And we also provide convenient hyperlinks to jump to other pages. In that way, you can get access to all relevant information through hyperlinks and go to any page you want. We will explain it later in detail to show you how convenient it is.

1.1.2 Basic Components of Homepage

The first part of our website is the home page. To build this page, we need to do the following things.

Navigation Bar On the top of our home page, we need a navigation bar to provide more convenient tools for users to use different functions of the website.

Search box The main part of the home page. Allow users to search whatever they want to know like Baidu and Google.

footer Show copyright information.

1.2 Navigation Bar

1.2.1 Sourcecode

```
<nav class="navbar navbar-default" role="navigation">
<div class="container-fluid">
<div class="navbar-header">
<a class="navbar-brand page-scroll" href="home.php"><img src="logo/logo_v4.png"
    width="110" height="30" alt="IEEE Search"></a>
</div>

<ul class="nav navbar-nav navbar-right">
<li><a href="pages/about_us.html"><span class="glyphicon glyphicon-user"></span>
    ?? </a></li>
</ul>
</div>
</nav>
```

1.2.2 Analysis

Navigation Bar On the top of our homepage, I use **<nav>** label to create a navigation bar. On the left side, I put a logo in ** **. Once you click the logo, you can return to homepage. On the right side, there is a button linked to about page.

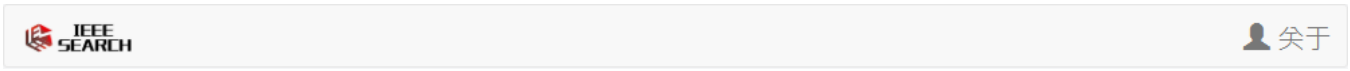


Figure 1.4: Navigation Bar

1.3 Header

1.3.1 Sourcecode

```
<header>
<div class="header-content">
<div class="header-content-inner">
<h1 id="homeHeading">IEEE <span class="text-colored">Search</span></h1>
<hr>
</div>
<div class="row">
<div class="tabbable" id="selection">
<div class="container nav-tab">
<ul class="nav nav-tabs">
```

```
<li class="active">
<a href="#home" data-toggle="tab">General</a>
</li>
<li class="">
<a href="#Author" data-toggle="tab">Author</a>
</li>
<li class="">
<a href="#Paper" data-toggle="tab">Paper</a>
</li>
<li class="">
<a href="#Affiliation" data-toggle="tab">Affiliation</a>
</li>
</ul>
</div>
<div class="tab-content">
<div class="tab-pane active" id="home">
<form class="form" action="result.php" method="get" name="searchform" id="searchForm" onsubmit="return checkform();">
<input type="text" class="search-wrapper form-control ui-autocomplete-input" type="text" id="key" name="key" placeholder="Author/Paper/Affiliation">
<button class="ace-search-style btn btn-circle" type="submit" value="Submit"><span class="glyphicon glyphicon-search" style="color:#ffffff" aria-hidden="true"></span><span class="btn-text">Search</span></button>
</form>
</div>
<div class="tab-pane" id="Paper">
<form class="form" action="pages/paper.php" method="get" name="searchform" id="searchForm" onsubmit="return checkform();">
<input type="text" class="search-wrapper form-control ui-autocomplete-input" type="text" id="PaperID" name="PaperID" placeholder="Paper">
<button class="ace-search-style btn btn-circle" type="submit" value="Submit"><span class="glyphicon glyphicon-search" style="color:#ffffff" aria-hidden="true"></span><span class="btn-text">Search</span></button>
</form>
</div>
<div class="tab-pane" id="Author">
<form class="form" action="pages/author.php" method="get" name="searchform" id="searchForm" onsubmit="return checkform();">
<input type="text" class="search-wrapper form-control ui-autocomplete-input" type="text" id="AuthorID" name="AuthorID" placeholder="Author">
<button class="ace-search-style btn btn-circle" type="submit" value="Submit"><span class="glyphicon glyphicon-search" style="color:#ffffff" aria-hidden="true"></span><span class="btn-text">Search</span></button>
</form>
</div>
<div class="tab-pane" id="Affiliation">
<form class="form" action="pages/affiliation.php" method="get" name="searchform" id="searchForm" onsubmit="return checkform();">
<input type="text" class="search-wrapper form-control ui-autocomplete-input" type="text" id="AffiliationID" name="AffiliationID" placeholder="Affiliation">
<button class="ace-search-style btn btn-circle" type="submit" value="Submit"><span class="glyphicon glyphicon-search" style="color:#ffffff" aria-hidden="true"></
```

```
    span><span class="btn-text">Search</span></button>
</form>
</div>
</div>
</div>

</div>
</div>
</header>

<div class="container footer" style="margin-top: -20px">
<p style="text-align: center">© IEEE 2018 Team 6</p>
</div>
</body>
</html>
```

1.3.2 Analysis

Nav-tab In homepage, I use **bootstrap** and **JavaScript** to create a navigation tab. The css pattern of the navigation bar is control by **"bootstrap.css"** and the JavaScript code to change the homepage content according to user's option. If the user click a tab with id "home" (for example), it will trigger a fucntion in **bootstrap.js** to change the input box into general search box.

Input Box Below the navigation tab, there's a input box for user to search whatever they're interested in. The input box and the submit button are created using bootstrap framework.

For each different option, there's a different input box with the according functions like atuocomplete, hint text and submit target destination. Take "General" option for example, the hint text is **"Author/Paper/Affiliation"** and submit button can confirm data transmission to **result.php**.

Formatting After we finish all the above elements we need in our homepapge, the last and important step for us is to format our page using **CSS (Cascading Style Sheets)** . The basic idea of fomatting our page is divide the homepage into different parts and fill in each part with different content. In **<html>**, we mainly use ** ** to represent each division and add property to it, reshape or beautify it in my way.

What needs to be addressed is I also add some property to the **autocomplete UI**.

```
<style>
        .ui-autocomplete {
        position: relative;
        max-width: 40vw;
        margin-left: 1vw;
        padding: 0.4vw 2.5vw 0.4vw 1.5vw;
        max-height: 300px;
        overflow-y: auto;
        overflow-x: hidden;
        font-size: 18px;
        }
</style>
```

max-width Control the max-width of the autocomplete box and thus make it not exceed the width of input box. (Otherwise it will look not so pleasant.)

max-height & overflow Make the autocomplete box show adequate information and not exceed the maximum height meanwhile.

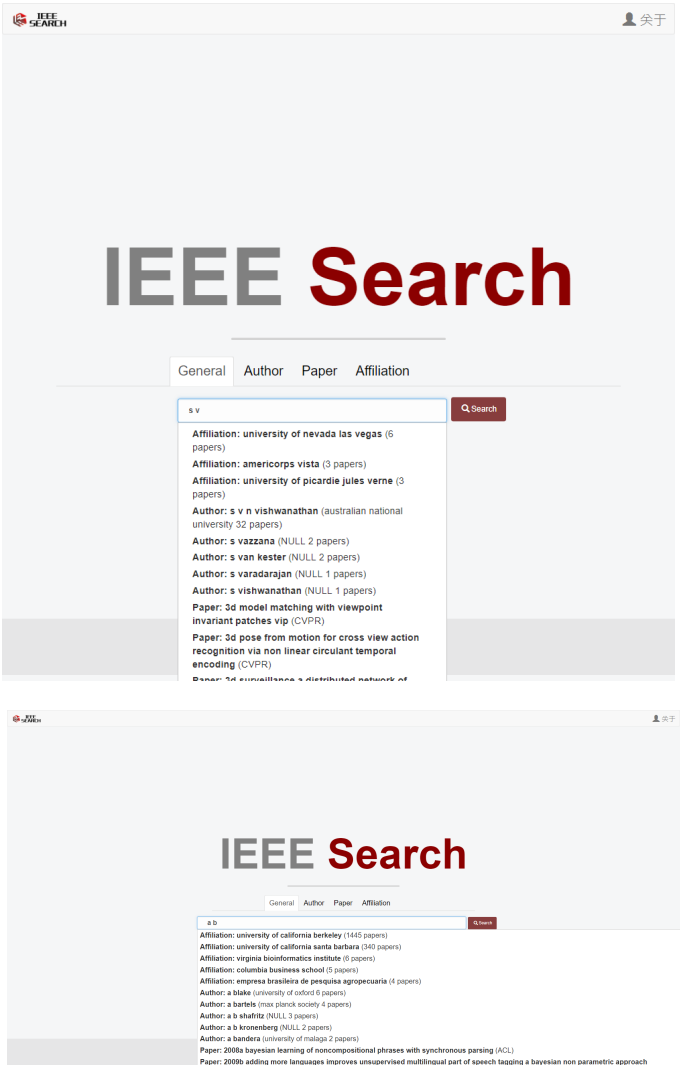


Figure 1.5: Some Situations Without CSS Adjustment to Autocomplete UI

See full css code in [home.css](#).

1.4 Autocomplete

Autocomplete When the user text in the input box, there will be a autocomplete box under the input box. The content of the autocomplete box will be determined by the user's option.

For example, when we use "General" option, the autocomplete box will be as the following image.

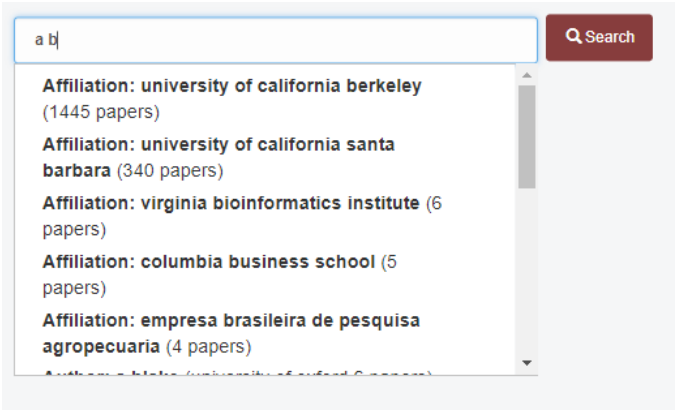


Figure 1.6: Autocomplete

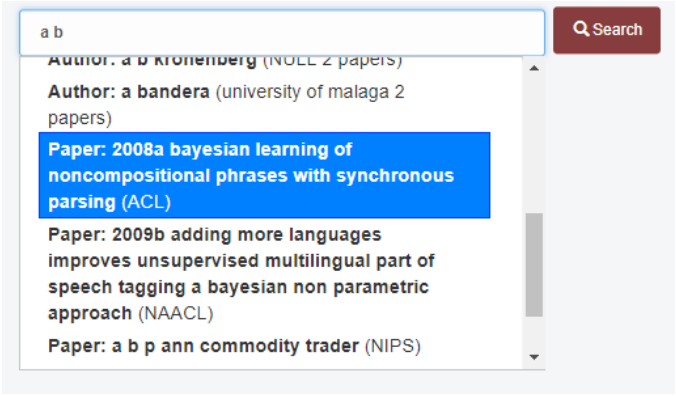


Figure 1.7: Autocomplete

As we can see, if we type something in the input box, the autocomplete function will list a series of possible result that the user want to search. In detail, we will label each piece of information with "Author", "Paper" or "Affiliation". Each labelled autocomplete information contains a hyperlink which links to the according page. For example, if you click the content with **Author** label, you will directly jump to **Author.php**

Next we will see how to implement a autocomplete widget

1.4.1 Sourcecode(cautocomplete.js)

```
1 $(function() {
2 $.widget( "custom.catcomplete", $.ui.autocomplete, {
3 _create: function() {
4 this._super();
5 this.widget().menu( "option", "items", "> :not(.ui-autocomplete-category)" );
```

```
},

_renderMenu: function( ul, items ) {
var that = this;

$.each( items, function( index, item ) {
var li = that._renderItemData( ul, item );
if ( item.category ){
li.attr( "aria-label",  item.label );
}
var div = li.children();
div.html('<strong>' + item.category + ': ' + item.value + '</strong> ' + item.info)
    ;
});
}
});

$("#key").catcomplete({
minLength: 2,
autofill: true,
source: function(request, response){
var host = document.location.href;
var ajaxurl = "http://localhost/hint.php";
$.ajax({
url: ajaxurl,
data: {
q: request.term
},
success: function(data){
var result = [];

var parseResult = JSON.parse(data);
var len = parseResult.aff.length;
for (var i = 0; i < len; ++i){
result.push({"URL": "http://localhost/pages/affiliation.php?AffiliationID=" +
    parseResult.aff[i].AffID, "label": parseResult.aff[i].label, "category": "
    Affiliation", "info": "(" + parseResult.aff[i].info + " papers)"})
}

len = parseResult.author.length;
for (var i = 0; i < len; ++i) {
result.push({"URL": "http://localhost/pages/author.php?AuthorID=" + parseResult.
    author[i].AuthorID, "label": parseResult.author[i].label, "category": "Author",
    "info": "(" + parseResult.author[i].aff + " " + parseResult.author[i].info + "
    papers)"})
}

len = parseResult.paper.length;
for (var i = 0; i < len; ++i) {
result.push({"URL": "http://localhost/pages/paper.php?PaperID=" + parseResult.paper
    [i].paperid, "label": parseResult.paper[i].label, "category": "Paper", "info": "
    (" + parseResult.paper[i].info + ")"})
```

```
}

response(result);
}
});
},
select: function( event, ui ){
window.location.href = ui.item.URL;
return false;
}
});

$("#AuthorID").catcomplete({
minLength: 2,
autofill: true,
source: function(request, response){
var host = document.location.href;
var ajaxurl = "http://localhost/hint_author.php";
$.ajax({
url: ajaxurl,
data: {
q: request.term
},
success: function(data){
var result = [];

var parseResult = JSON.parse(data);

var len = parseResult.author.length;
for (var i = 0; i < len; ++i) {
result.push({"URL": "http://localhost/pages/author.php?AuthorID=" + parseResult.
    author[i].AuthorID, "label": "Name: " + parseResult.author[i].label+" ID: "+
    parseResult.author[i].AuthorID, "category": "Author", "info": "(" + parseResult.
    author[i].aff + " " + parseResult.author[i].info + " papers)"})
}

response(result);
}
});
},
select: function( event, ui ){
window.location.href = ui.item.URL;
return false;
}
});

$("#PaperID").catcomplete({
minLength: 2,
autofill: true,
source: function(request, response){
var host = document.location.href;
var ajaxurl = "http://localhost/hint_paper.php";
```

```
$.ajax({
url: ajaxurl,
data: {
q: request.term
},
success: function(data){
var result = [];

var parseResult = JSON.parse(data);

var len = parseResult.paper.length;
for (var i = 0; i < len; ++i) {
result.push({"URL": "http://localhost/pages/paper.php?PaperID=" + parseResult.paper
    [i].paperid, "label": "ID: " + parseResult.paper[i].paperid+" Title: "+
    parseResult.paper[i].label, "category": "Paper", "info": "(" + parseResult.paper
    [i].info + ")"})
}

response(result);
}
});
},
select: function( event, ui ){
window.location.href = ui.item.URL;
return false;
}
});

$("#AffiliationID").catcomplete({
minLength: 2,
autofill: true,
source: function(request, response){
var host = document.location.href;
var ajaxurl = "http://localhost/hint_affiliation.php";
$.ajax({
url: ajaxurl,
data: {
q: request.term
},
success: function(data){
var result = [];

var parseResult = JSON.parse(data);

var len = parseResult.aff.length;
for (var i = 0; i < len; ++i){
result.push({"URL": "http://localhost/pages/affiliation.php?AffiliationID=" +
    parseResult.aff[i].AffID, "label": "Name: " + parseResult.aff[i].label + " ID: "
     + parseResult.aff[i].AffID, "category": "Affiliation", "info": "(" +
    parseResult.aff[i].info + " papers)"})
}

```

```
response(result);
}
});
},
select: function( event, ui ){
window.location.href = ui.item.URL;
return false;
}
});

});
```

1.4.2 Analysis

cautocomplete.js

Basic Structure The basic structure of **cautocomplete.js** is shown in the following graph.

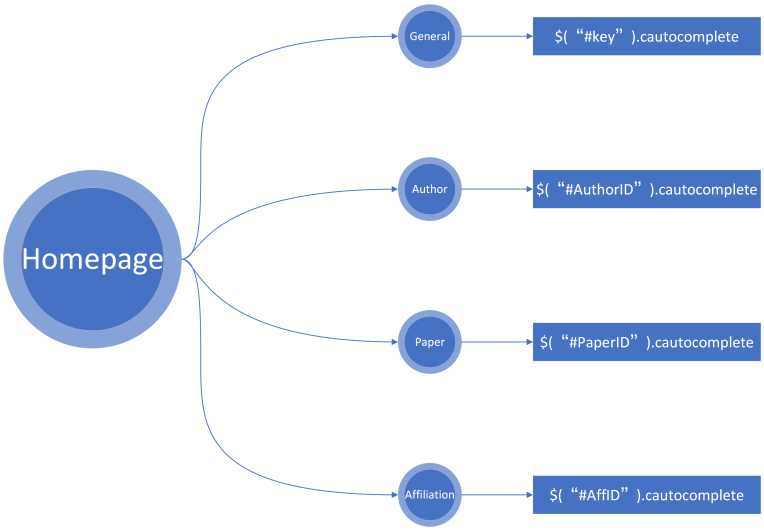


Figure 1.8: Autocomplete Structure

As is shown above, different opitons in **hompage** will call different functions in **cautocomplete.js**.

Ajax To realize the autocomplete function, **Ajax** is the most essential part.

In the **jQuery** function, we use $.ajax() function to sent an ajax request to target url (which is, in our website, hint.php). If the request is success, it will execute a function to process the data.

```
success: function(data){
        function                # data processing
}
```

And the data processing method isn't hard to understand. That is:

1. Transfer the **JSON** type data into a **JavaScript** array. (JSON.parse(data))

2. Design a for loop in **JavaScript** language to extract every piece of information and add them into a new array result. In this array, there are four features of each element:

- **URL** The destination url
- **label** Main information (e.g. Authorname, AuthorID)
- **category** The type of this information ("Author", "Paper" or "Affiliation")
- **info** Some additive information. (e.g. Paper number of this author, Affiliation)

jQuery UI Autocomplete is a widget in **jQuery UI**, all we have to do is extend the function and design one autocomplete widget based on the original one.

Widget UI With function $.widget("custom.autocomplete", $.ui.autocomplete) we can format the content displayed on the autocomplete widget UI. For each information, we put it in a **<li>**. And we use **** label to address some content. (item.label and item.value).

```
$.each( items, function( index, item ) {
        var li = that._renderItemData( ul, item );
        if ( item.category ){
                li.attr( "aria-label",  item.label );
        }
        var div = li.children();
        div.html('<strong>' + item.category + ': ' + item.value + '</strong> ' +
           item.info);
});
```

Autocomplete Function With function $(id).cautocomplete(), we can set some feature to control the autocomplete widget.

1. minLength: The minimum length that will trigger autocomplete function
2. source: Autocomplete data source
3. select:function() the function to be called when the user click one item in the select box.

hint.php

SQL Query In **hint.php** there're several SQL search queries. With like *a* and limitin sql , we can do fuzzy search to find related information in the database.

json_encode() After we get our result and reshape it into the right form, we can use **json_encode()** to submit the data into the server and thus be caught by **cautocomplete.js**

Notice The json data that we submit should have the same index name and variable name as it is in the autocomplete function, otherwise it may cause an error.

Take Author information for example.

```
$author[] = array(
'AuthorID' => $row['AuthorID'],
```

```
'label' => $row['AuthorName'],
'info' => $row['papernum'],
'aff' => $row1['AFFILIATIONNAME']
);
```

1.5 Improvements

1.5.1 SQL Search Speed

To increse the search (autocomplete) speed, I tried the following ways:

1. Replace INNER JOIN with LEFT JOIN

2. Add Index to each table

3. Setting the maximum row number of the result. (for a autocomplete box, we only show top 5 authors, top 5 affiliations and top 5 papers)

hint.php?q=a%20	200	xhr	jquery-3.1.1.min.js:4	1.6 KB	675 ms
hint.php?q=a%20b	200	xhr	jquery-3.1.1.min.js:4	1.6 KB	144 ms
hint.php?q=a%20b	200	xhr	jquery-3.1.1.min.js:4	1.6 KB	161 ms
hint.php?q=a%20b	200	xhr	jquery-3.1.1.min.js:4	1.6 KB	178 ms
hint.php?q=a%20	200	xhr	jquery-3.1.1.min.js:4	1.6 KB	601 ms
hint.php?q=a%20c	200	xhr	jquery-3.1.1.min.js:4	1.7 KB	157 ms

Figure 1.9: Autocomplete

As it can be seen in the picture, the speed of autocomplete function is relatively high.

Possible Improvement In the next version of our IEEE Search website, we may apply **elasticsearch** in it and make the SQL search faster.

1.5.2 Apperance Of The Webpage

Before designing the apperance of the webpage, it looked like this:

Figure 1.10: Autocomplete

There's nothing but a input box and a submit button. But after I add css to it, it looks like this:

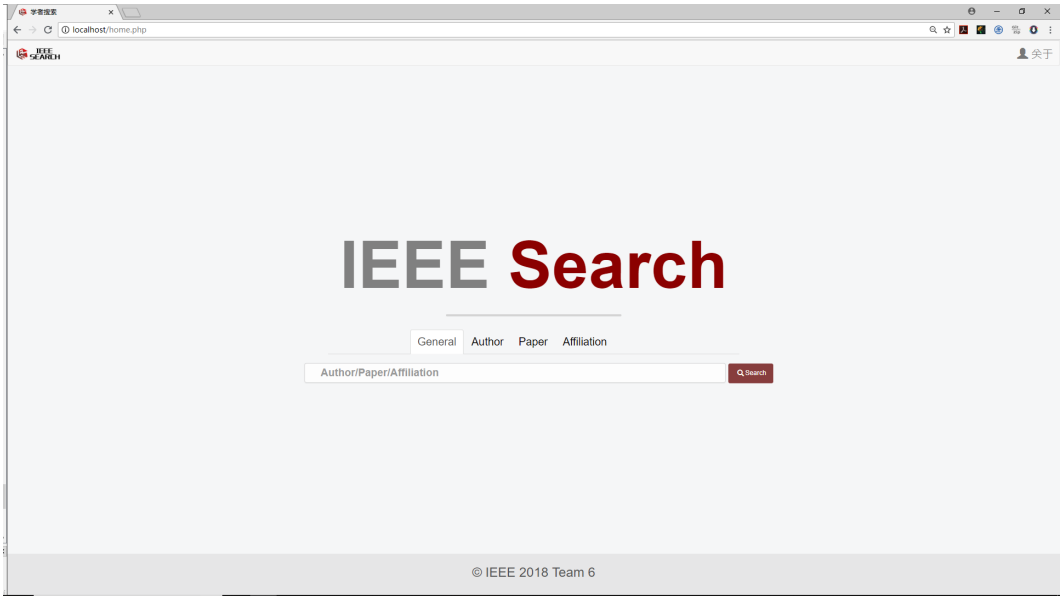


Figure 1.11: Autocomplete

Possible Improvement Maybe there're more features we can add to the homepage. Like **Recommandations**, **Latest News** and **Ranking list of authors and papers** etc.

Chapter 2

Result Display

2.1 Description

2.1.1 Webpage Display

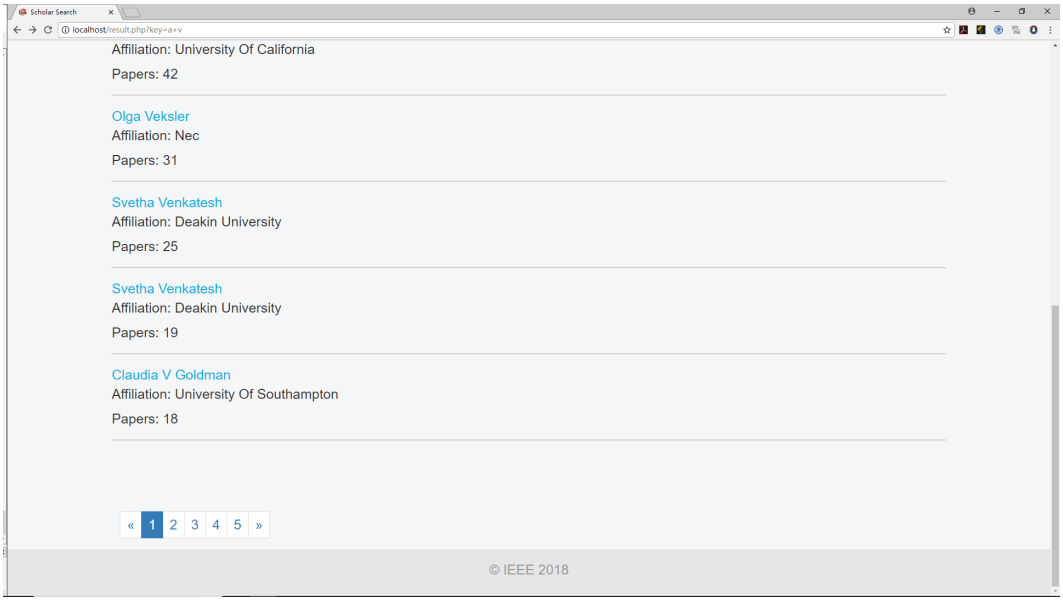


Figure 2.1: Result Display Page

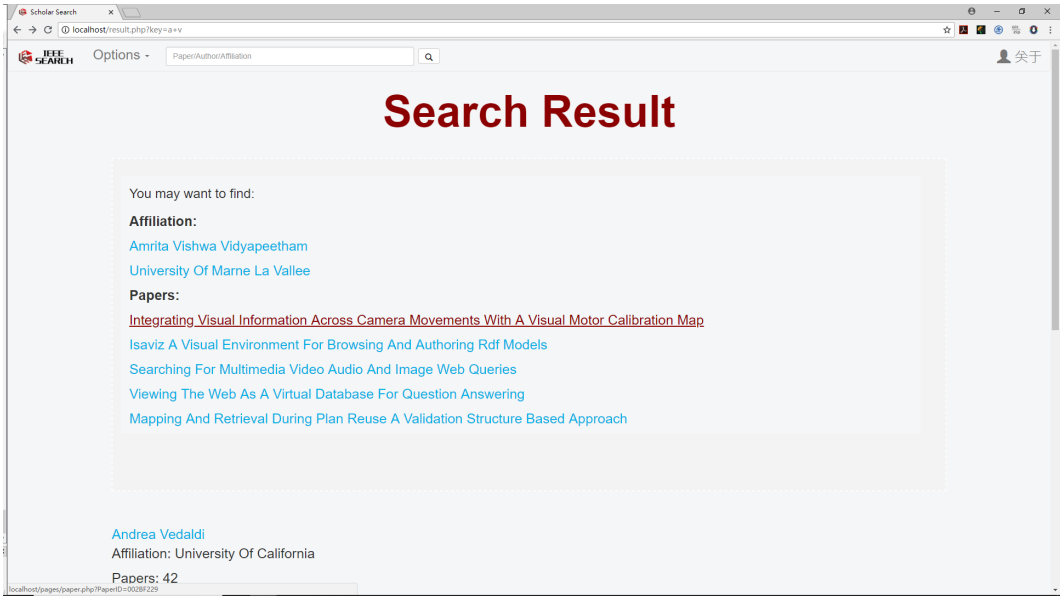


Figure 2.2: Result Display Page

As is shown in the above picture, the result page consists of 4 parts: navigation bar, recommendation box, author result and pagination.

Navigation Bar Compared with the navigation bar in the homepage, the navigation bar in the result display pgae has an additional search bar on the top of the page which has exactly the same function as the homepage does. It means we won't have to go back to homepage to search.

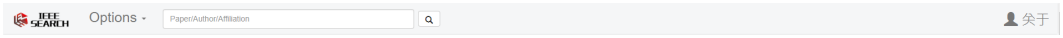


Figure 2.3: Navigation of Result Page

Recommendation Box After the navigation bar and the header is the recommendation box. The recommendation information is based on user's input. Take "ab" for example, the webpage will give some recommendation results that contatin word "ab" and list them under the label **Affiliation** or **Papers**.

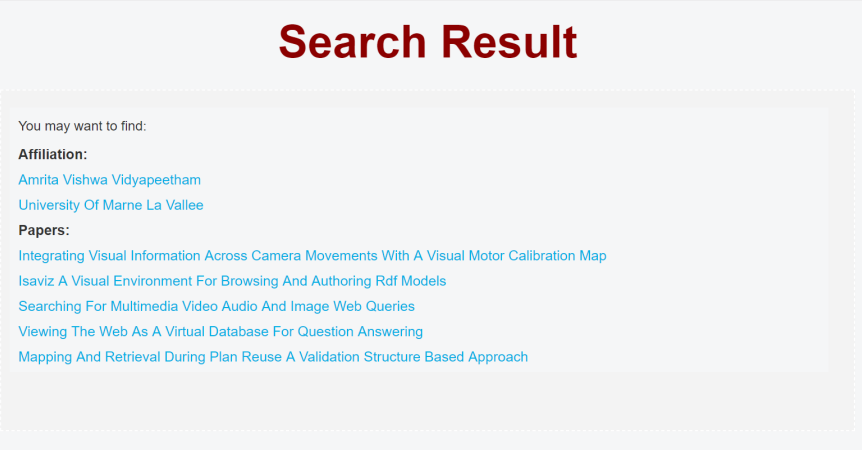
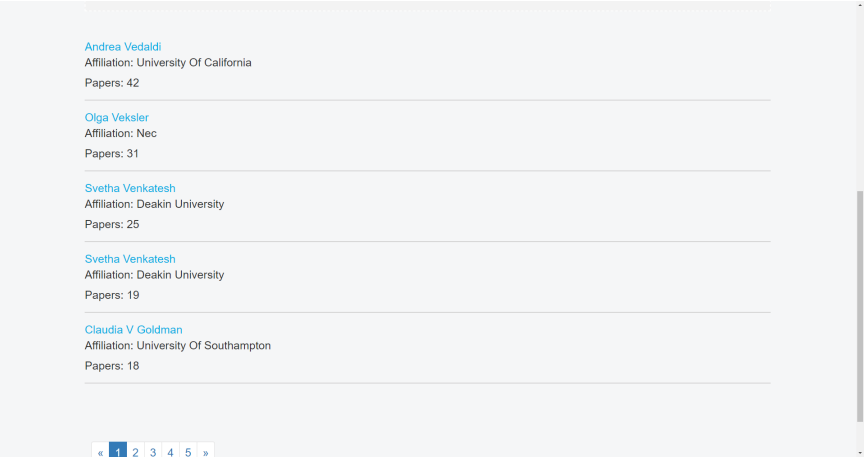
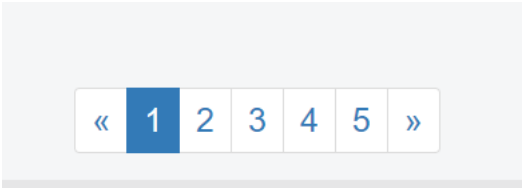


Figure 2.4: Recommendation

Author Result All search result about author name will be display in the area below the recommendation box. For each author, it will display their name, major affiliation and paper number.



Pagination In the bottom of result page there's a pagination division. Using pagination buttons, user can switch to the particular page, go to previous page and go to next page.



2.2 Navitaion Bar

2.2.1 Sourcecode

```
<nav class="navbar navbar-default" role="navigation">
<div class="container-fluid">
<div class="navbar-header">
<a class="navbar-brand page-scroll" href="home.php"><img src="logo/logo_v4.png"
    width="110" height="30" alt="IEEE Search"></a>
</div>
<ul class="nav navbar-nav">
<li class="dropdown">
<a href="#" class="dropdown-toggle" data-toggle="dropdown">
Options <b class="caret"></b>
</a>
<ul class="dropdown-menu">
<li>
<a href="javascript:author()">Author</a>
</li>
<li>
<a href="javascript:paper()">Paper</a>
</li>
<li>
<a href="javascript:affiliation()">Affiliation</a>
</li>
<li class="divider"></li>
<li>
<a href="home.php">Home</a>
</li>
<li class="divider"></li>
<li>
<a href="#">Contact Us</a>
</li>
</ul>
</li>
</ul>
<form id="navsearch" action="result.php" method="get" class="navbar-form navbar-
    left" role="search" style="width:50%">
<div class="form-group" style="width:50%">
<input type="text" id="key" name="key" class="search-wrapper form-control ui-
    autocomplete-input"
placeholder="Paper/Author/Affiliation" style="width:100%">
</div>
<button type="submit" class="btn btn-default"><span class="glyphicon glyphicon-
    search"></span></button>
</form>
<ul class="nav navbar-nav navbar-right">
<li><a href="#"><span class="glyphicon glyphicon-user"></span>  </a></li>
</ul>
```

2.2.2 Analysis

Dropdown Menu The dropdown menu has replaced the navigation tab in the homepage to offer different option for user to select (General, Affiliation, Paper). Clicking one of the option will click the according **JavaScript** function in result.js.

```
function paper() {
        $("#navsearch").attr("action", "pages/paper.php");
        $("input").attr("placeholder", "Paper");
        $("input").attr("id", "PaperID");
        $("input").attr("name", "PaperID");
        }
```

In the above code, I mainly use **jQuery** selector to change the properties (**href, placeholder, id and name**).

Search Box The search box is different with the one in homepage. Beacuse it's linked to different pages (**Author.php, Paper.php or Affiliation.php**). But each input box has autocomplete function, too.

2.3 Recommendation Box

2.3.1 Sourcecode: Back-end

```
$sug1 = mysql_query("SELECT AUTHORNAME FROM AUTHORS1 WHERE AUTHORNAME LIKE '%{$input}%' LIMIT 5");
$sug2 = mysql_query("SELECT AFFILIATIONNAME, AFFILIATIONID FROM AFFILIATIONS WHERE AFFILIATIONNAME LIKE '%{$input}%' LIMIT 5");
$sug3 = mysql_query("SELECT TITLE, PAPERID FROM PAPERS WHERE TITLE LIKE '%{$input}%' LIMIT 5");
$sug4 = mysql_query("SELECT CONFERENCENAME, CONFERENCEID FROM CONFERENCES WHERE CONFERENCENAME LIKE '%{$input}%' LIMIT 3");

$affiliation = null;
$title = null;
$conference = null;
while ($row = mysql_fetch_array($sug2))

$affiliation[] = array("affiliation"=>$row["AFFILIATIONNAME"], "id"=>$row['AFFILIATIONID']);

while ($row = mysql_fetch_array($sug3))
$title[] = array("title"=>$row["TITLE"], "id"=>$row["PAPERID"]);

while ($row = mysql_fetch_array($sug4))
$conference[] = array("name" => $row['CONFERENCENAME'], "id"=>$row["CONFERENCEID"]);
```

2.3.2 Analysis

Sql Search Search for any matching **conference, affiliation and paper** according to the key word that's converted to result.php. In the above php codes, I use three simple search statement to fetch search result. After fetching result, I put them in three arrays: **$affiliation, $title and $conference**.

Data Transmission In the end, we use **echo json_encode()** to convert the recommendation result to the font-end.

```
echo json_encode(array("status"=>$status, "affiliation"=>$affiliation, "title"=>
    $title, "result"=>$info, "PageSize"=>$PageSize, "PageNum"=>$PageNum, "conference
    "=>$conference));
```

Display Recommdation (result.js) After transmitting recommendation data, I use **JavaScript** statement in result.js to rearrange them into a js array. Then I use **document.getElementById()** function to put new <html> element into result.php.

```
if (result.conference != null) var str3 = "<p><strong>Conference:</strong></p>";
for (var c in result.conference) {
        var conf = result.conference[c].id;
        str3 += "<p><a href='pages/conference.php?ConferenceID=" + conf + "' style
            ='text-transform: uppercase'>" + result.conference[c].name + "</a>" + "
            </p>";
}
// Take paper recommendation as an example
```

```
document.getElementById('suggestion').innerHTML = str1 + str2 + str3;
```

```
<div class="container-fluid" id="related-result" style="padding: 30px;">
        <div class="container-fluid row">
                <h3>You may want to find: </h3>
                <div class="column" id="suggestion">
                </div>
        </div>
</div>
<!-- After Js function, we will add data into the division with the id "suggestion"
    -->
```

2.4 Author Result

2.4.1 Sourcecode

get_result.php Similarily, we can get author result with Sql search and php.

```
# $result is the array containing all author result.
$len = count($result);

$PageSize = 5;
$PageNum = ceil($len/$PageSize);
```

```
6
7  $info = null;
8  if ($affiliation == null && $title == null && $conference == null && $result ==
       null) $status = 0;
9  else $status=1;
10 if ($status == 1){
11         foreach ($result as $id => $value):
12         {
13                 if ($id >= $pagenum * $PageSize && $id < ($pagenum + 1) * $PageSize
                       ) {
14                 $info[] = array("Papers" => $value[1], "AuthorName" => utf8_encode(
                       $value[0]), "Affiliation" => $value[2], "AuthorID" => $value[3])
                       ;
15                 }
16         }
17         endforeach;
18 }
```

result.js JavaScript code in result.js.

```
1  $.getJSON("get_result.php?page=" + current + "&key=" + key, function (result) {
2          var str = '';
3          for (var i in result.result) {
4                  var info = result.result[i];
5                  str += "<a href='pages/author.php?AuthorID=" +
6          info.AuthorID + "' class='authorname'>" + info.AuthorName + "</a><p style='
               text-transform: capitalize'>Affiliation: " + info.Affiliation + "</p>" +
                "<p>Papers: " + info.Papers + "</p><hr class='minline'>";
7          }
8  }
9  document.getElementById('result').innerHTML = str;
```

result.php The container in result.php.

```
1  <div class="container-fluid" style="margin-top: 3%">
2          <div class="row" id="result"></div>
3  </div>
```

2.4.2 Analysis

$.getJSON(url, function(success)) With this **jQuery** function, I get the JSON data transmitted by get_result.php.

CSS Each line is put in a **<p>** label. And after each author, there's a **<hr>** to seperate the data.

Transmission According to the page value, the back-end php code will return different data. To ensure the transmitted data is divided into pages, I use a **foreach** loop to visit every result in result array and use an **if** statement to judge whether the result is in this page.

```
1          if ($id >= $pagenum * $PageSize && $id < ($pagenum + 1) * $PageSize)
```

Then we can return the results in the particular page the user click.

2.5 Pagination

2.5.1 Sourcecode

Back-end In get_result.php.

```
$len = count($result);
$PageSize = 5;
$PageNum = ceil($len/$PageSize);
```

Front-end result.php & result.js

When first enter **result.php**, there's a initialization of pagination.

```
pagination = "<ul class=\"pagination\"><li><a href='javascript:pre()'>&laquo;</a></
    li>";

if (pagenum > totalpage) {
        pagination += "<li id=\"" + '0' + "\" class=\"active\"><a href=\"javascript
            :switchpage(0)\">" + 1 + "</a></li>";
        for (var i = 1; i < totalpage; i++) {
        var pageindex = parseInt(i) + 1;
        pagination += "<li id=\"" + i + "\"><a href=\"javascript:switchpage(" + i +
             ")\">" + pageindex + "</a></li>";
        }
}

else {
        pagination += "<li id=\"" + '0' + "\" class=\"active\"><a href=\"javascript
            :switchpage(0)\">" + 1 + "</a></li>";
        for (var i = 1; i < pagenum; i++) {
        var pageindex = parseInt(i) + 1;
        pagination += "<li id=\"" + i + "\"><a href=\"javascript:switchpage(" + i +
             ")\">" + pageindex + "</a></li>";
        }
}
```

Pagination Switch page with buttons with numbers on them.

```
function switchpage(pageindex) {
        $("#"+current).removeClass("active");
        if (pageindex < pagenum && pageindex >= 0)
        $.getJSON("get_result.php?page="+pageindex+"&key="+key, function(result,
            status){
        var str = '';
        for(var i in result.result)
        {
        var info = result.result[i];
```

```
        str += "<a href='http://localhost/pages/author.php?AuthorID="+
        info.AuthorID+"' class='authorname'>"+info.AuthorName+"</a><p style='text-
            transform: capitalize'>Affiliation: "+info.Affiliation+"</p>"+"<p>Papers
            : "+info.Papers+"</p><hr class='minline'>";

        }
        document.getElementById('result').innerHTML=str;
        });
        current = pageindex;

        if ((current >= 2) && (current < pagenum - 2)) adjust();
        $("#"+current).addClass("active");
}
```

Pagination Switch page with previous button or next button.

```
function pre() {
        $("#"+current).removeClass("active");
        current--;

        if (current < pagenum && current >= 0)
        {$.getJSON("get_result.php?page="+current+"&key="+key, function(result,
            status){
        var str = '';
        for(var i in result.result)
        {
        var info = result.result[i];
        str += "<a href='http://localhost/pages/author.php?AuthorID="+
        info.AuthorID+"' class='authorname'>"+info.AuthorName+"</a><p style='text-
            transform: capitalize'>Affiliation: "+info.Affiliation+"</p>"+"<p>Papers
            : "+info.Papers+"</p><hr class='minline'>";

        }
        document.getElementById('result').innerHTML=str;
        });

        }
        else { alert("No more! "); current++;}

        if ((current >= 2) && (current < pagenum - 2)) adjust();
        $("#"+current).addClass("active");
}


function next() {
        $("#"+current).removeClass("active");
        current++;
        if (current < pagenum)
        {$.getJSON("get_result.php?page="+current+"&key="+key, function(result,
            status){
        var str = '';
```

```
        for(var i in result.result)
        {
        var info = result.result[i];
        str += "<a href='http://localhost/pages/author.php?AuthorID="+
        info.AuthorID+"' class='authorname'>"+info.AuthorName+"</a><p style='text-
            transform: capitalize'>Affiliation: "+info.Affiliation+"</p>"+"<p>Papers
            : "+info.Papers+"</p><hr class='minline'>";

        }
        document.getElementById('result').innerHTML=str;
        });

        }
        else { alert("No more! "); current--;}

        if ((current >= 2) && (current < pagenum - 2)) adjust();
        $("#"+current).addClass("active");
}
```

Adjust Adjust buttons when the page number is changed.

```
function adjust() {
        pagination = "<ul class=\"pagination\"><li><a href='javascript:pre()'>&
            laquo;</a></li>";

        for (var i=current-2; i<current+3; i++) {
        var pageindex = parseInt(i) + 1;
        pagination += "<li id=\"" + i + "\"><a href=\"javascript:switchpage(" + i +
             ")\">" + pageindex + "</a></li>";
        }

        pagination += "<li><a href='javascript:next()'>&raquo;</a></li></ul>";
        document.getElementById('page').innerHTML=pagination;
        $("#"+current).addClass("active");
}
```

2.5.2 Analysis

Initialization After getting the first batch of data from back-end page, we should intialize the pagination buttons on the bottom of result display page. We use **<ul class="pagination">** to create a **bootstrap** style pagination and add hyperlinks to each **<li>**. For each button of pagination, it's linked to a JavaScript function (switchpage(), pre() and next()). At last, we add **class="active"** to the button represents page 1.

switchpage(pageindex) Change page number into **pageindex + 1**

1. Use $(#id).removeClass("active") to remove the "active" class of the current page.
2. Check if the new page number is valid.
3. Repeat the same process in initialization process to get author result in current page number.
4. The last step is to change the pagination buttons if necessary and add class "active" to the new page number.

```
if ((current >= 2) && (current < pagenum - 2)) adjust();
$("#"+current).addClass("active");
```

adjust() The adjust function is relatively easy. I just repeat the word in initialization similarliy.

```
for (var i=current-2; i<current+3; i++) {
        var pageindex = parseInt(i) + 1;
        pagination += "<li id=\"" + i + "\"><a href=\"javascript:switchpage(" + i +
            ")\">" + pageindex + "</a></li>";
 }
```

The above codes are important to use a for loop to make a new pagination.

Data Visualization(The tree and pie chart)
(Zhibang Wang)

Chapter 3

Echart

I use echarts as the tool to help me made this charts.

3.1 Introduction

Echart is a declarative framework for rapid construction of web-based visualization made of baidu. It provide many kinds of different charts.
They contribute ECharts, an easy-to-use framework to construct interactive visualization. The main contribution sconfirm to three goals.
Easy-to-use. There are some difficulties for users to learn the visual representations if a declarative language is employed. It is desirable to allow users to focus on the design of the visualization rather than on the use of some tools.
Rich builit-in interactions. Efficient data exploration and analysis demand a wealth of configurable interactions. ECharts designs and implements rich built-in interactions that are attached to each chart type, minimizing therequirement of customization of user.
High performance. By introducing a streaming system architecture andincremental rendering mode, high performance is achieved with ECharts, even when handling millions of data points.

3.2 How It Work

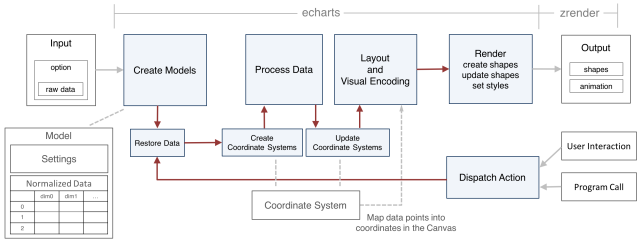


Figure 3.1: Echart

The echarts provide a data interface so that we can put specific data in it. Then it will transform the data to a model which echarts can show. After some work, it can display in the website with animation. Also, it is strong

interactive, which mean that you can do some opeartion to the chart and it will refresh the data or transport the message you need. The picture show the principle clearly.

Chapter 4

The tree charts

4.1 Introduction

We use the tree charts to show the author's tutors or student.The tree can be expanded if you click it. So we have to tree to show the two relationship. Cause the two tree is designed in the same method, I just show how I made the teacher tree in this chapter and the student tree code is similiar as it and you can see it in the complete code.
The data type of the tree:[**"name":name,"id":id,children:[....],"value":0**], the chidlren include his teachers' data.

4.2 The Difficulty

Firstly, I want to get the whole tree data and push them in echart so that the tree can be finished. I want to use recursion function to geet the author's teachers and their teachers... until the people have no teacher in the database. However, this method failure because the size of the data is too large and it always exceed the limit. And after I made the tree in the other method, I found that the recursion is difficult to end. Also, lots of time and room is wasted because we don't want to expand all the tree. So I try other method.
Finally, I try to create the tree in a dynamic method. That is, we first initial the tree, and if we click the tree, it get the new data and refresh the tree so that it will be expanded.

4.3 Solution Design

The problem can be divided in two part: **initial the tree** and **expand the tree**.

4.3.1 Initialization

First, we should get the data of the author's toturs from the database. Because we reach the author page use the GET method, so we already know his information. Using his AuthorID,we can get the data.

```
1 $sql = "SELECT AuthorName FROM authors WHERE AuthorID = \"".$id."\"";
2
3 $result = mysqli_query($conn, "set names utf8");
4 $result = mysqli_query($conn, $sql);
5 $row = mysqli_fetch_array($result);
```

```
$name = $row['AuthorName'];
echo $row['AuthorName']."</h1>";
$children=array();
$teacher= mysqli_query($conn, "select cooperation.TutorID,authors.AuthorName from
    cooperation inner join authors on authors.AuthorID=cooperation.TutorID where
    cooperation.StudentID='".$id."' and cooperation.Bool=1");
while($row=mysqli_fetch_array($teacher))
{
$arr = array('name' =>ucwords($row['AuthorName']),'id'=>$row['TutorID'],'value'=>0
    );
$children[]=$arr;
};
/**/$tree_root=array('name'=>ucwords($name),'id'=>$id,'children'=>$children,'value
    '=>0);
$jstree = json_encode($tree_root);// □□□□
file_put_contents('tree.json', $jstree);
```

So we write the data in the tree.json as soon as we open a author's page. And I use a **iframe** to show the tree in the author page. The tree chart is in the my tree.php. It can put the data in the tree.json into the chart model so that the tree can be initial. Using getjson to do this work.

```
$.getJSON('tree.json', function (data) {
option.series[0].data.push(data);
require.config({
paths: {
echarts: 'http://echarts.baidu.com/build/dist'
}
});
require(['echarts', 'echarts/chart/tree'], function(ec) {
chart = ec.init($("#main")[0]);
chart.clear();
chart.setOption(option);
var ecConfig = require('echarts/config');
chart.on(ecConfig.EVENT.CLICK, clickFun2);

})
});
```

Push the data in the option.series. And I set the option to make the color, size of the chart.

```
option = {
tooltip: {
trigger: 'item',
formatter: '{b}:{c}',
hideDelay: 0 ,
animation:true,
animationDurationUpdate: 750,
},
series: [{
name: 'tree',
type: 'tree',
orient: 'horizontal', // vertical horizontal
rootLocation: { x: '10%', y: '20%' }, // □□□□□  {x: 'center',y: 10}
```

```
nodePadding: 20, ////////////////////////////////////////
symbol: 'circle',
symbolSize: 4,
roam: true,
symbolSize: 7,
//direction: 'inverse',
label: {
normal: {
position: 'left',
verticalAlign: 'middle',
align: 'right',
fontSize: 9
}
},

leaves: {
label: {
normal: {
position: 'right',
verticalAlign: 'middle',
align: 'left'
}
}
},
expandAndCollapse: true,
animationDuration: 550,
animationDurationUpdate: 750,
data: []
}]
};
```

The data is empty before we get from tree.json.
Then the tree is initial and you can see it in the page.

4.3.2 Expansion

The effect we want to achieve is that when we click the point, it can get data from the database to the chart data. And then refresh the chart so it seem that the tree is expand. I want to use echarts4.0 to do this work, but failed to do it because i can not alert the leave's data. I use echart2.0 to do this work.
Echarts provide us to make events to do such kind of work. It is the most important part of expansion.

```
function clickFun2(param) {
// console.log(JSON.stringify(param));
if(!(param.data.children && param.data.children.length > 0)) {
console.log('open');
if(param.data.children_bak) {
param.data.children = param.data.children_bak;
}
else {
var x=param.data.id;
console.log(x);
chart.showLoading();
```

```
$.getJSON("creat_tree.php?author="+x,function(result){
param.data.children=result;
console.log(param.data.children);
chart.hideLoading();
chart.refresh();
})
}
} else {
console.log('close');
param.data.children_bak = param.data.children;
param.data.children = [];
}
//console.log(param);
//chart.setOption(option);
chart.refresh();
```

If the leave have children, then we move the children to anothor place so that the his children points can be hidden. If the leave have not children, the function will jugde whether they are hidden. If they are moved, the function move them back so that the leave is expanded again. If there are no children, it use getjson to get json from createtree.php, in which another function can get the children data.
The function is:

```
function createTree($id)
{
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "main_db";

$conn = mysqli_connect($servername, $username, $password, $dbname);
//$conn = new mysqli($servername, $username, $password, $dbname);

if (!$conn) {
die("Connection failed: " . mysqli_connect_error());
}
$children=array();
//$teacher= mysqli_query("select teacher from relation where student='".$id."' and judge=1");
$teacher= mysqli_query($conn,"select cooperation.TutorID,authors.AuthorName from cooperation inner join authors on authors.AuthorID=cooperation.TutorID where cooperation.StudentID='".$_GET["author"]."' and cooperation.Bool=1");
while($row=mysqli_fetch_array($teacher))
{
//$arr = array('name' =>$row['teacher'],'value'=>0 );
$arr = array('name' =>ucwords($row['AuthorName']),'id'=>$row['TutorID'],'value'=>0 );
$children[]=$arr;
//$children[]=createTree($row['teacher'],$n-1);
};
return $children;
}
```

It is similiar as the initializationz:Search all the co-workers of the author and if they are his teacher then get their

id and name. We transport the id to the function and show the result in this:

```
$all=createTree($_GET["author"]);
$json_string = json_encode($all);
echo $json_string;
```

So the tree is completed.

4.4 Exhibition

4.4.1 The Database

In lab4, we train the model and made the predicted totur-student relation table, this is the basic of the tree charts.

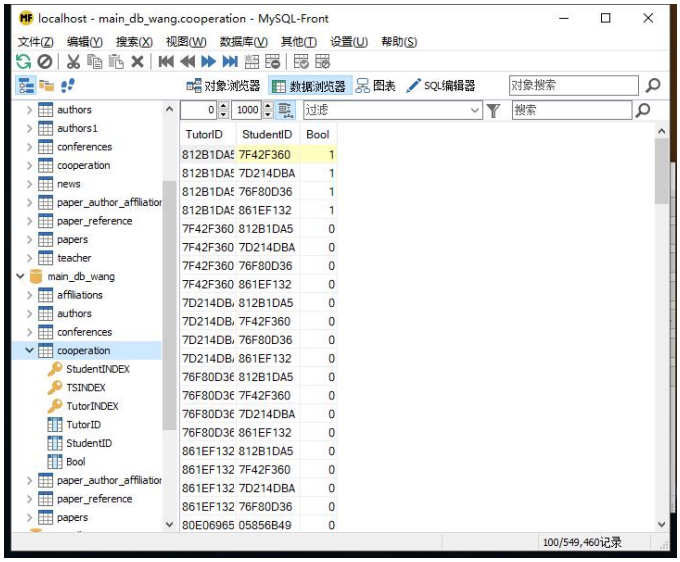


Figure 4.1: Database

4.4.2 The Initialized Tree

In the author page, we use iframes to display the charts whicj is showed in tree.php.

4.4.3 The Expanded Tree

If we click the tree, it will be expand and it will be shrinked after re-click.

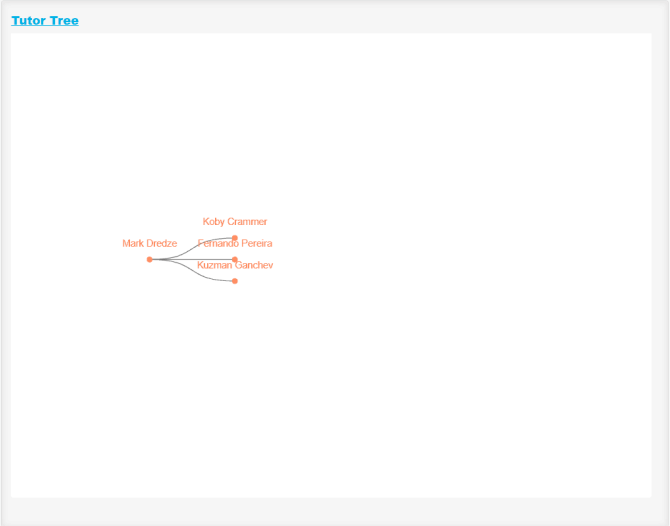


Figure 4.2: Tutor Tree

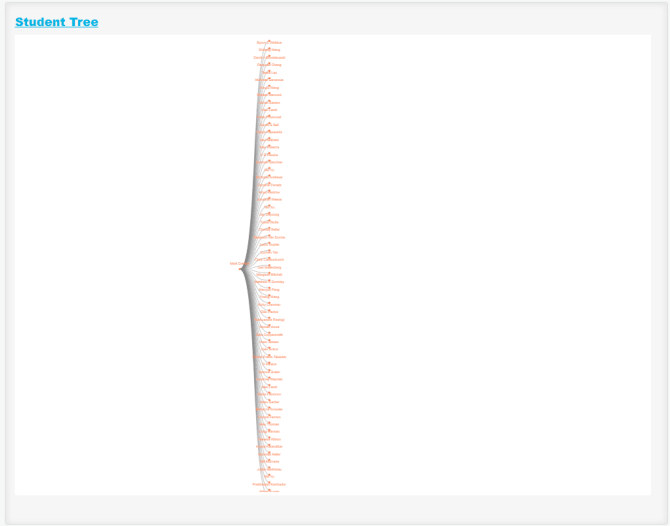


Figure 4.3: Student Tree

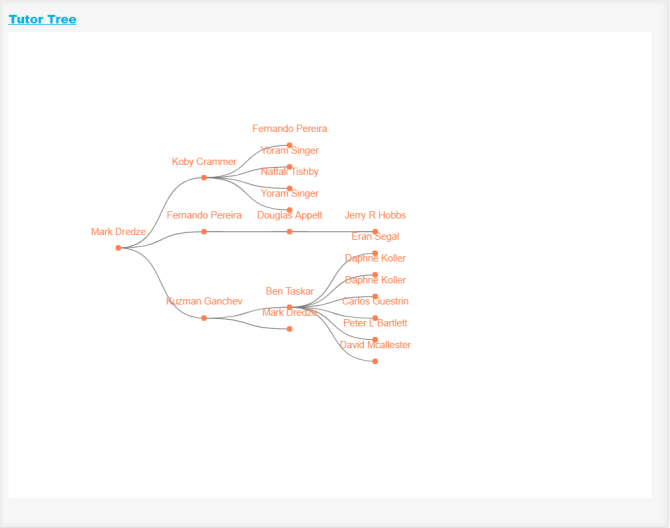


Figure 4.4: Tutor Tree

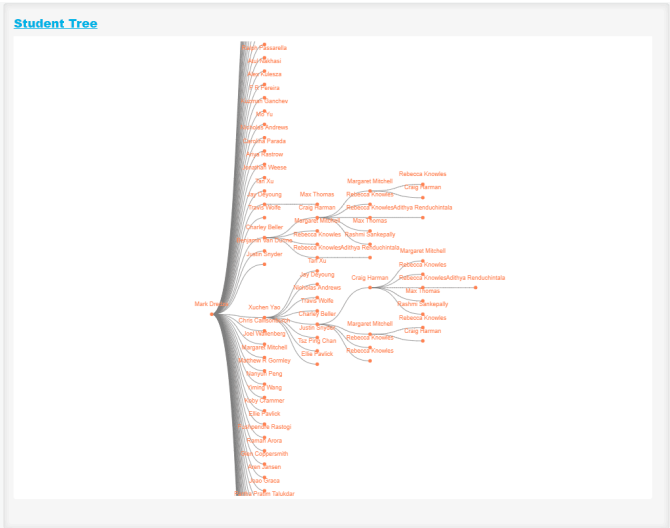


Figure 4.5: Student Tree

Chapter 5

Pie chart

5.1 Introduction

Create a pie chart to show the author's papers distribution in different conference.
Because I also use Echarts to solve this problem, so some step is the same. But the data type is different: **('value':the number of the conference's appearance,'name':conference name,'id':conference id)**. And this graph don't need to be expanded.

5.2 Solution Design

This time I use Echarts4.0 because it have better animation.
Firstly, we should set the opion and get the data. The getJSON is a good way:

```
var author="<?php echo $_GET["author"]; ?>";
$.getJSON("create_pie.php?author="+author,function(result){
console.log(result);
myChart.setOption({
title: {
text: 'Venue Published In',
//subtext: □□□□'',
left: 'center'
},
tooltip : {
trigger: 'item',
formatter: "{b} : {c} ({d}%)"
},
legend: {
// orient: 'vertical',
// top: 'middle',
bottom: 10,
left: 'center',
data: result[0],
},
series : [
{
```

```
type: 'pie',
radius : '65%',
center: ['50%', '50%'],
selectedMode: 'single',
data:result[1],
itemStyle: {
emphasis: {
shadowBlur: 10,
shadowOffsetX: 0,
shadowColor: 'rgba(0, 0, 0, 0.5)'
}
}
}
]
});
```

Get the AuthorID from php and use getJSON to get the data from the createpie.php, which can get the data we need after connect the sql.

```
$conference= mysqli_query($conn,"select papers.ConferenceID,conferences.
    ConferenceName,count(papers.ConferenceID) from paper_author_affiliation inner
    join papers on paper_author_affiliation.paperID=papers.paperID left join
    conferences on conferences.ConferenceID=papers.ConferenceID where
    paper_author_affiliation.authorID='".$_GET["author"]."' group by papers.
    ConferenceID");
$result1=array();
$result2=array();
$arr=array();
while ($row=mysqli_fetch_array($conference)) {
$result1[]=$row['ConferenceName'];
$a=array('value'=>$row['count(papers.ConferenceID)'],'name'=>$row['ConferenceName
    '],'id'=>$row['ConferenceID']);
$result2[]=$a;
}
$arr[]=$result1;
$arr[]=$result2;
```

result1 is the title legend of the graph,showing what conference appeared. result2 is the data of the graph, showing the rate of each conference.
Then we can create the pie.

```
myChart.showLoading();
$.getJSON
.......
......
myChart.hideLoading();
});;
if (option && typeof option === "object") {
myChart.setOption(option, true);
};
```

Before we get the data successfully, the graph will show "loading"(But in fact it get data very fast and you can seldom see the "loading").

What's more, I add a double-click event on it. If you double click it, the conference page will be opened.

```
myChart.on('dblclick', function (params) {
console.log(params.data.id);
window.open('conference.php?ConferenceID=' + encodeURIComponent(params.data.id));
});
```

5.3 Exhibition

The pie display different conferences in different color. You can move your cursor on it and it will show the name and the rate.Click it to highlight this part.

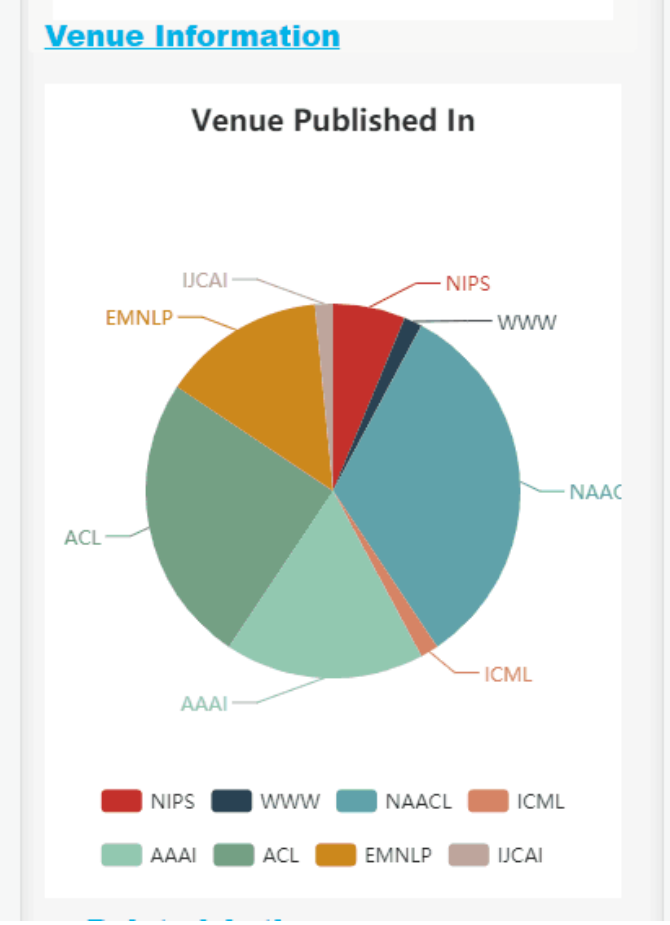


Figure 5.1: Pie Graph

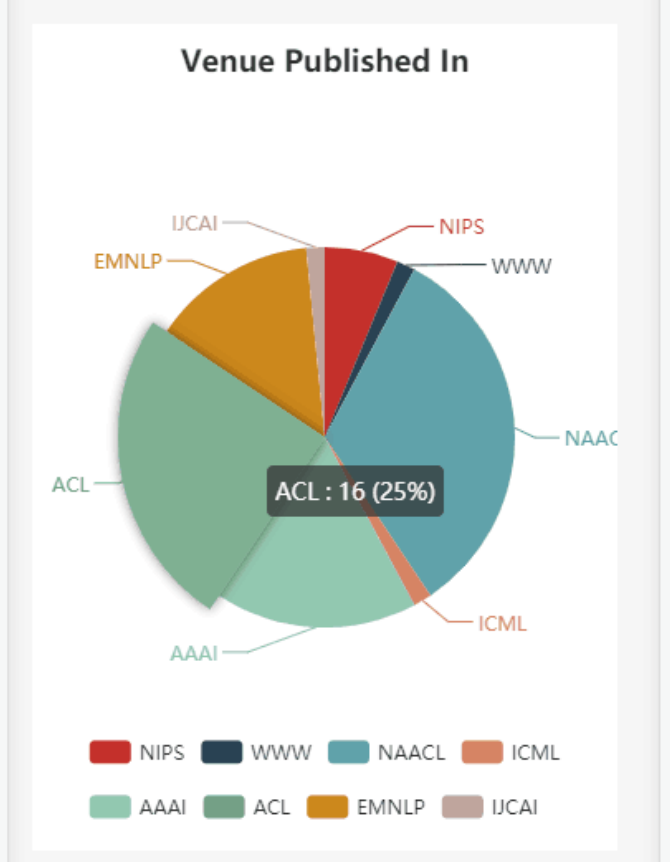


Figure 5.2: The name and rate

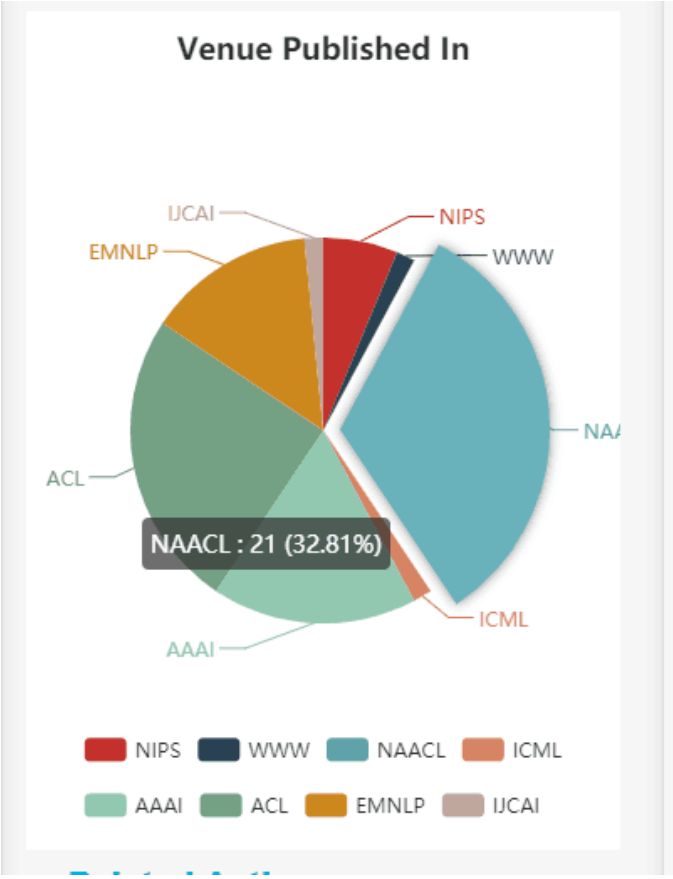


Figure 5.3: Highlight Part

Figure 5.4: DBclick to open the conference page

AuthorPage/PaperPage/ AffiliationPage/ConferencePage (Weidong Wang)

Chapter 6

Author Page

6.1 Elements in Author Page

Author Page will show the information about this author, such as:

- Author fundamental information.
- The papers published by the author.
- The time line-chart about the author's papers.
- The pie-chart about the conferences the author published papers in.
- Two tree graph about the author's teachers and his/her students.
- The cooperation relationship graph about the author and his/her cooperator.
- The related authors.

6.2 Author Infomation

Some information such as the Author's name, affiliations and conferences would be shown.

6.2.1 Author Name

It's easy to show the name with the AuthorID from the method "get".

```
1         SELECT AuthorName FROM authors WHERE AuthorID = \"".$_GET["AuthorID"]."\""
```

6.2.2 Author's Conferences

1. Find all the conferences the author published papers in.

```
1                   SELECT ConferenceID FROM (
2                   SELECT PaperID FROM paper_author_affiliation
3                   WHERE AuthorID = \"".$_GET['AuthorID']."\") AS TA
```

```
                LEFT JOIN papers ON TA.PaperID=papers.PaperID
                GROUP BY ConferenceID ORDER BY count(ConferenceID) DESC
```

2. Link every ConferenceID into ConferenceName.

```
                SELECT ConferenceName FROM conferences
                WHERE ConferenceID=\"{$row['ConferenceID']}\"
```

3. Show all the conferences in the website linked with comma.

```
                $af = "";
                while ($row = mysqli_fetch_array($result)) {
                $sql = "SELECT ConferenceName FROM conferences
                WHERE ConferenceID=\"{$row['ConferenceID']}\"";
                $result1 = mysqli_query($conn, "set names utf8");
                $result1 = mysqli_query($conn, $sql);
                $row1 = mysqli_fetch_array($result1);
                $af = $af."<a href=\"http://localhost/conference.php".
                "?ConferenceID={$row['ConferenceID']}\" ".
                "target=\"_blank\">".$row1['ConferenceName']."</a>, ";
                }
                if ($af != "") echo chop($af, ", ");
                else echo "None";
```

4. The final string: "ECCV, CVPR, ICCV, NIPS, ICML"

6.2.3 Author's Affiliations

1. It's similar to author's conferences.

2. Find all the AffiliationID about the Author except "None".

```
                SELECT AffiliationID FROM paper_author_affiliation
                WHERE AuthorID=\"".$_GET["AuthorID"]."\"
                GROUP BY AffiliationID
                ORDER BY count(AffiliationID) DESC
```

3. Link every AffiliationID into AffiliationName.

```
                SELECT AffiliationName FROM affiliations
                WHERE AffiliationID=\"{$row['AffiliationID']}\"
```

4. Show all the affiliations in the website linked with comma.

```
                $af = "";
                while ($row = mysqli_fetch_array($result)) {
                if ($row['AffiliationID'] != 'None') {
                $sql = "SELECT AffiliationName FROM affiliations
                WHERE AffiliationID=\"{$row['AffiliationID']}\"";
                $result1 = mysqli_query($conn, "set names utf8");
                $result1 = mysqli_query($conn, $sql);
                $row1 = mysqli_fetch_array($result1);
```

```
                $af = $af."<a href=\"http://localhost/affiliation.php".
                "?AffiliationID={$row['AffiliationID']}\" ".
                "target=\"_blank\">".$row1['AffiliationName']."</a>, ";
                }
                }
                if ($af != "") echo chop($af, ", ");
                else echo "None";
```

6.3 Author's papers

1. The element shows all the papers the author wrote. And the title, conferences, affiliations and citations are shown.
2. Only 10 papers are shown. And you can click buttons to change the other papers by JavaScript.
3. The amount of papers and the current and final PageNumber are shown in this elements.
4. The shown way is similar to Experiment IV.

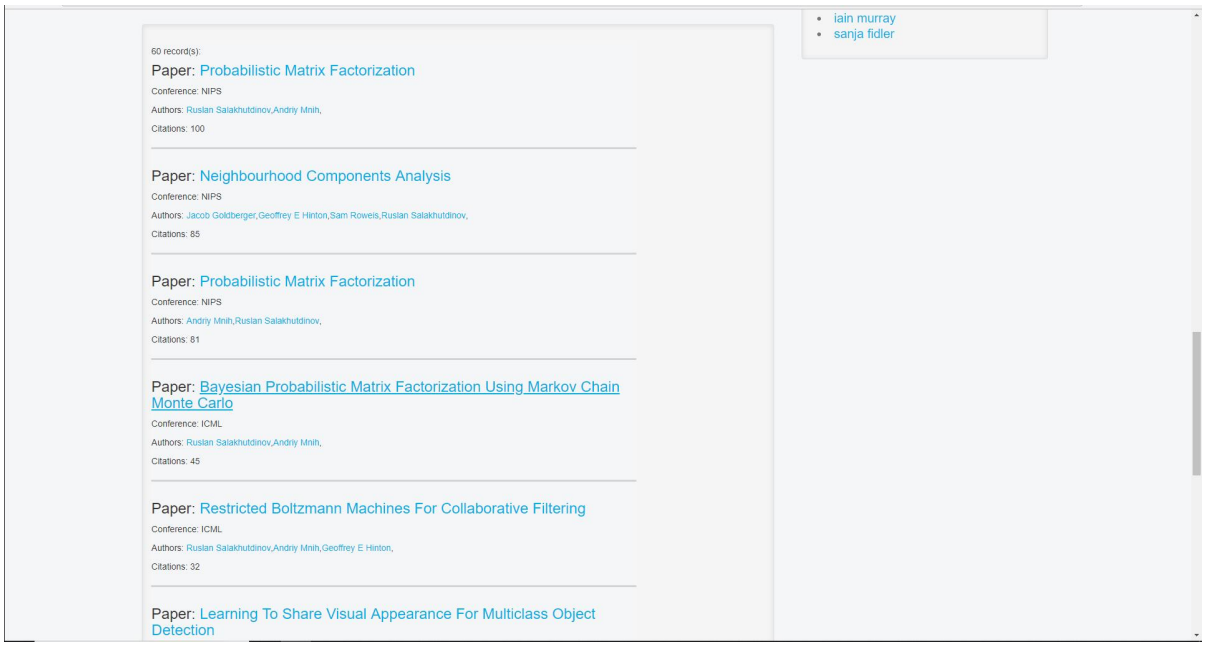


Figure 6.1: The author's papers.

6.4 The Cooperation Relationship Graph

1. The teacher-student relationship has been found and saved in the Experiment IV.
   - Using Python to find every two-author arrangement. Save the data.
   - Using Python to find every two authors' cooperation feature. Save the data.

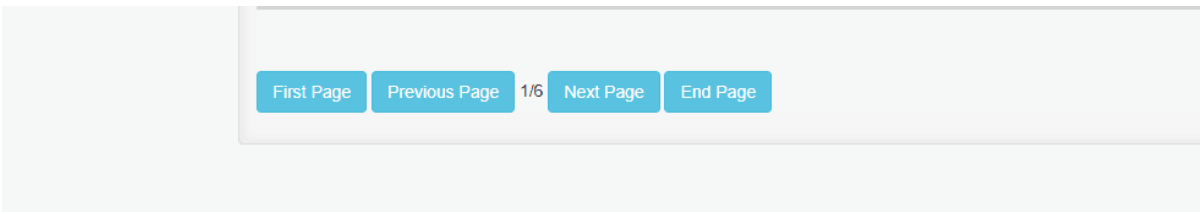


Figure 6.2: The papers' buttons.

- Using the trained classifier in Experiment III to predict the data.(Using the LogisticRegression of the library "sklearn".)
- Saving the bool value about the relationship of two scholars.

2. Find the current author's cooperators by searching every papers he published.
3. check if the cooperator is the teacher or the student. Give different labels about them(teacher, student or both not).
4. find every two cooperators' cooperation times. Using it as the value of the cooperation link.
5. Draw a framework about "D3.js" in the author page.
6. Using the method AJAX and the function "getJSON" of "jQuery.js", fill the framework.

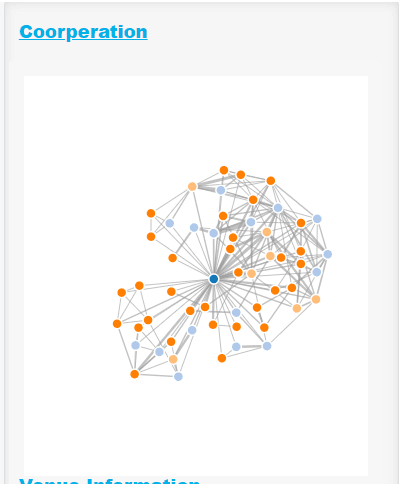


Figure 6.3: The author's cooperation relationship graph.

6.5 Related Authors

1. The recommendation depends on the cooperation times.
2. Firstly, use a framework from "bootstrap.css" to show the related Authors.

```
1                <div class="similar">
2                <div class="author-contact">
```

```
                <div style="height:10px; overflow:hidden"></div>
                <span class="glyphicon glyphicon-user"></span>
                <strong> Related Authors</strong>
                <div class="text-muted">
                <div style="height:6px; overflow:hidden"></div>
                <?php ... ?>
                </div>
                </div>
                </div>
```

3. Find all the papers the author wrote.

```
                SELECT PaperID FROM paper_author_affiliation
                WHERE AuthorID=\"{$_GET["AuthorID"]}\"
```

4. Find all the cooperators.

```
                foreach ($papers as $value) {
                $sql = "SELECT TA.AuthorID, AuthorName FROM
                (SELECT AuthorID FROM paper_author_affiliation
                WHERE PaperID = \"{$value}\") AS TA
                LEFT JOIN Authors ON TA.AuthorID = Authors.AuthorID";
                $result = mysqli_query($conn, "set names utf8");
                $result = mysqli_query($conn, $sql);
                while (($row=mysqli_fetch_array($result))) {
                if (!in_array($row['AuthorID'], $authors) &&
                strcmp($row['AuthorID'], $_GET["AuthorID"]) != 0) {
                array_push($authors, $row['AuthorID']);
                }
                }
                }
```

5. Calculate the cooperation times. Using insert-sort to get the top-10 authors. Show on the web page.

```
                $coop = array();

                for ($j = 1; $j < $aulen; ++$j) {
                $conum = count(array_intersect(
                $authorpaper[0], $authorpaper[$j]));
                if ($conum != 0)
                $coop[] = array(
                "target" => $authors[$j],
                "value" => $conum
                );
                }

                unset($authorpaper);

                $aulen--;
                $temp = array();
                for ($i = 0; $i < $aulen; ++$i) {
                $temp = $coop[$i];
```

```
19                  for ($j = $i-1;
20                  $j >= 0 && $temp["value"] > $coop[$j]["value"]; --$j) {
21                  $coop[$j+1] = $coop[$j];
22                  }
23                  $coop[$j+1] = $temp;
24                  }
25
26                  $coop = array_slice($coop, 0, 10);
27
28                  foreach ($coop as $key) {
29                  $sql = "SELECT AuthorName FROM authors
30                  WHERE AuthorID = \"{$key['target']}\"";
31                  $result = mysqli_query($conn, "set names utf8");
32                  $result = mysqli_query($conn, $sql);
33                  while (($row=mysqli_fetch_array($result))) {
34                  echo "<li style=\"padding-left:10px\">";
35                  echo "<a href=\"author.php?AuthorID=".
36                  "{$key['target']}\">{$row['AuthorName']}</a>";
37                  echo "</li>";
38                  }
39                  }
```

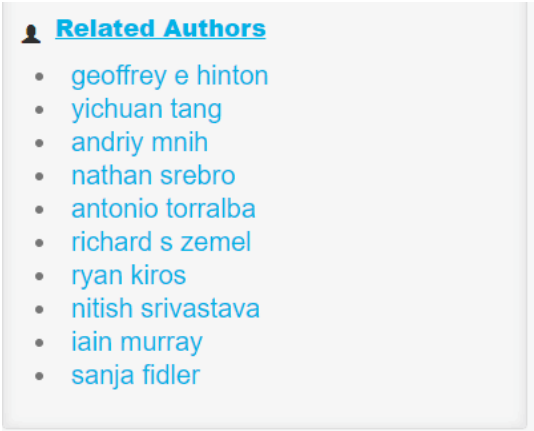


Figure 6.4: The related authors.

6.6 The Final Page

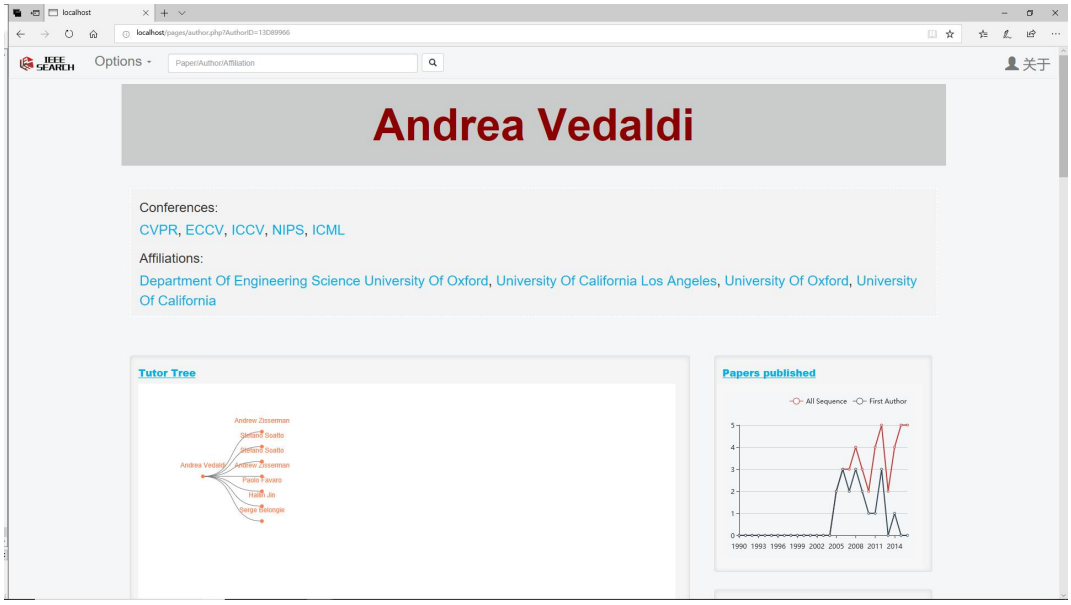


Figure 6.5: The Author page (1).

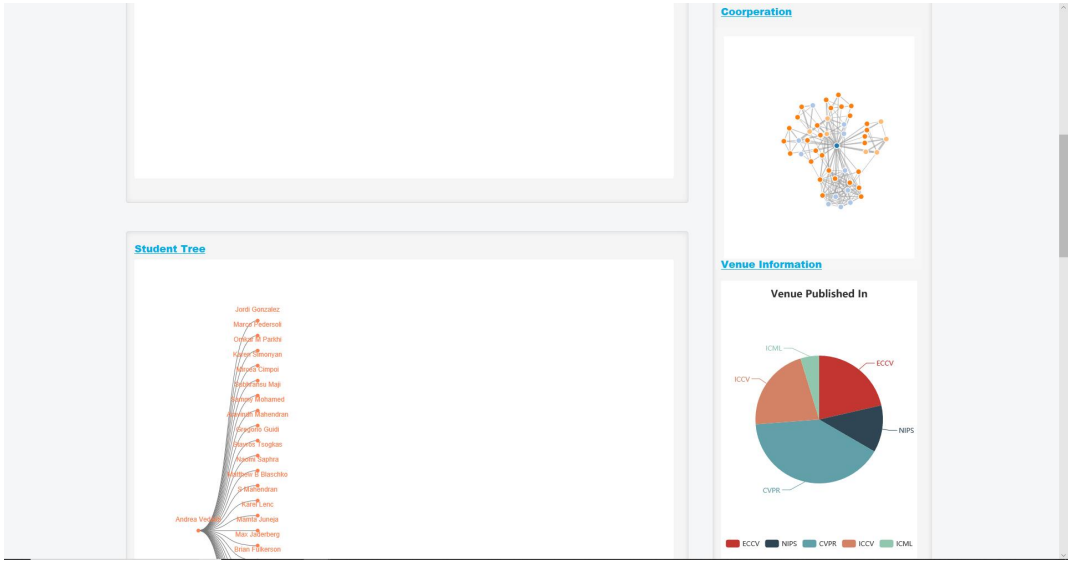


Figure 6.6: The Author page (2).

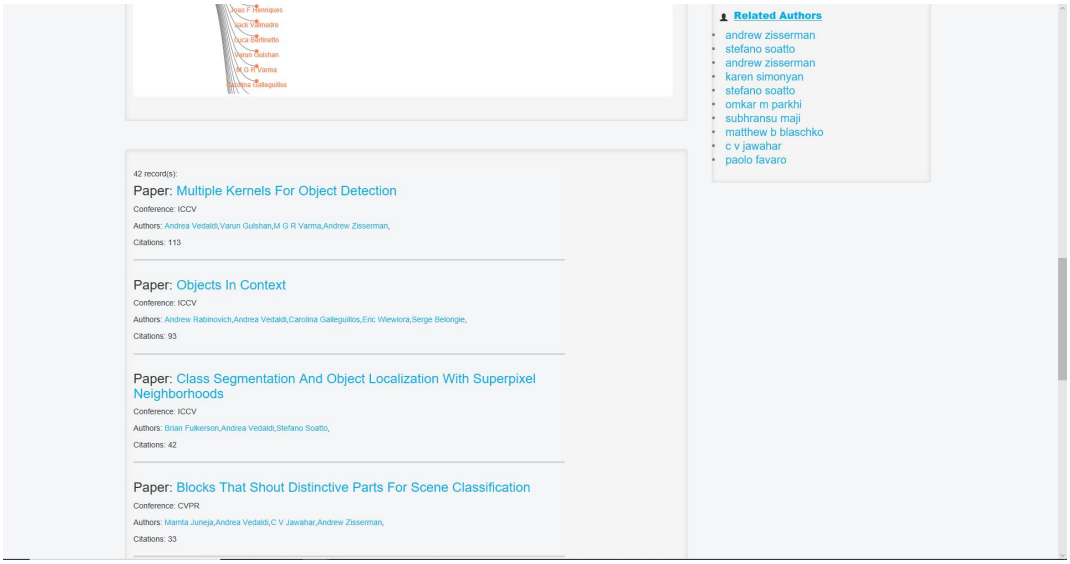


Figure 6.7: The Author page (3).

Chapter 7

Paper Page

7.1 Elements in Paper Page

Paper Page will show the information about this paper, such as:

- The fundamental information about this paper.
- The reference information about this paper.
- The citations information about this paper.
- The recommended papers about this paper.

7.2 Navigation Bar

Different informations show in different blocks. Use a framework from "bootstrap.css" to achieve it.

```
1        <div class="container-fluid">
2        <div class="container display">
3        <ul class="nav nav-tabs" role="tablist">
4        <li role="presentation" class="active">
5        <a href="#info" role="tab" data-toggle="tab">Info</a>
6        </li>
7        <li role="presentation">
8        <a href="#references" role="tab" data-toggle="tab">
9        References</a>
10       </li>
11       <li role="presentation">
12       <a href="#citations" role="tab" data-toggle="tab">
13       Citations</a>
14       </li>
15       <li role="presentation">
16       <a href="#similar" role="tab" data-toggle="tab">
17       Similar</a>
18       </li>
19       </ul>
20       </div>
```

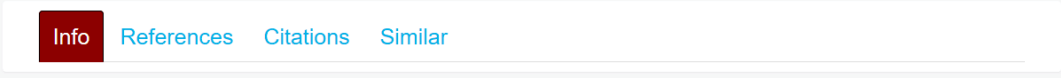


Figure 7.1: The paper navigation bar.

7.3 Paper Information

Fundamental information includes the title, the published year, the conference published in and the affiliation about the authors.

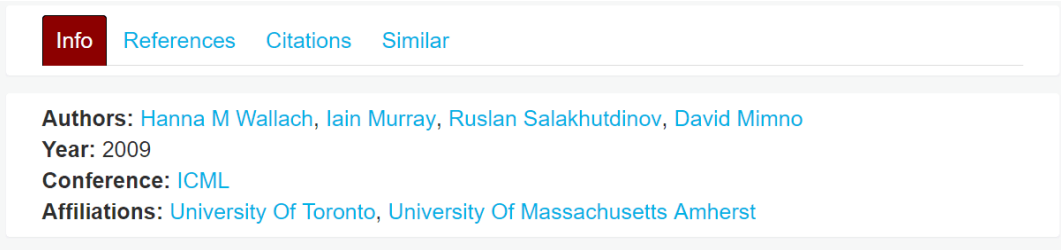


Figure 7.2: The paper's fundamental information.

7.4 References

This part shows all the papers this paper referred to. They were arranged in accordance with the referred time. Using the method AJAX and the function "getJSON" of "jQuery.js", find the data and show them in the page. Every page has 10 papers, which can use buttons to change. The way is just like the author page.

```
        SELECT Title, TD.PaperID, PaperPublishYear,
        ConferenceID, Referred FROM
        (SELECT TC.PaperID, Title, PaperPublishYear,ConferenceID
        FROM (SELECT referenceID as PaperID FROM paper_reference
        WHERE PaperID=\"{$_GET["PaperID"]}\") AS TC
        JOIN papers ON TC.PaperID=papers.PaperID) AS TD
        LEFT JOIN (SELECT referenceID, count(referenceID) as Referred
        FROM paper_reference GROUP BY referenceID) AS TA
        ON TD.PaperID=TA.referenceID ORDER BY Referred DESC limit i, 10
```

```
        while ($row = mysqli_fetch_array($result)) {
        ...
        $au = "";
        while ($row1 = mysqli_fetch_array($result1)) {
        $au = $au."<a href=\"http://localhost/pages/author.php".
        "?AuthorID={$row1['AuthorID']}\"".
        " target=\"_blank\">{$row1['AuthorName']}</a><br>";
        }

        $data2[] = array(
```

```
        'title' => "<a href=\"http://localhost/pages/paper.php?".
        "PaperID={$row['PaperID']}\">{$row['Title']}</a>",
        'conference' => $row2['ConferenceName'],
        'year' => (string)($row['PaperPublishYear']),
        'authors' => $au,
        'referred' => $ref
        );
        }
        echo json_encode($data2);
```

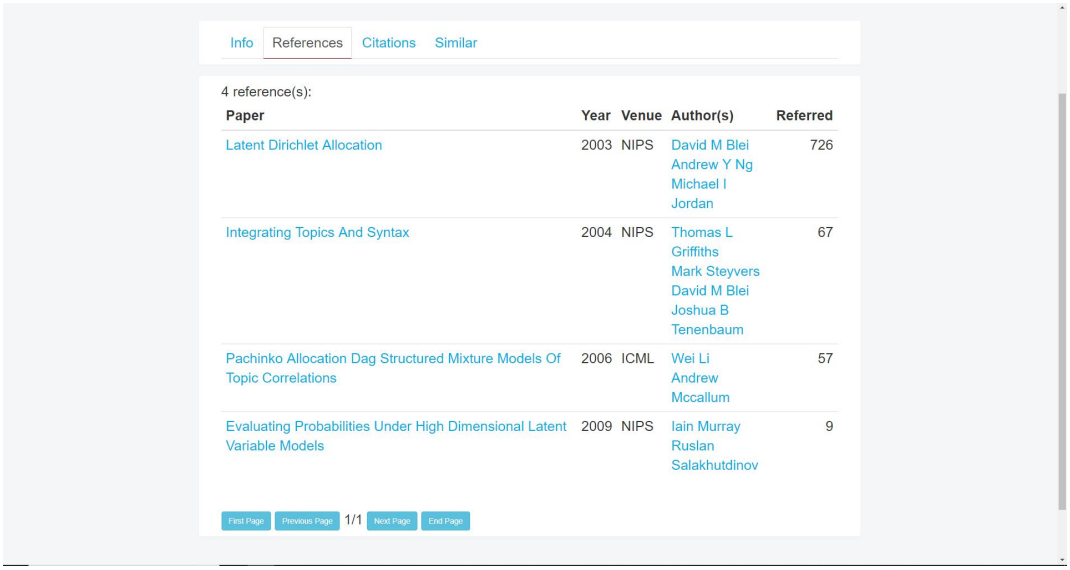


Figure 7.3: The paper's reference information.

7.5 Citations

The citations page is just like the reference page, only the papers are other papers referred to this paper.

```
        SELECT Title, TD.PaperID, PaperPublishYear,
        ConferenceID, Referred FROM
        (SELECT TC.PaperID, Title, PaperPublishYear,ConferenceID
        FROM (SELECT PaperID FROM paper_reference
        WHERE referenceID=\"{$_GET["PaperID"]}\") AS TC
        JOIN papers ON TC.PaperID=papers.PaperID) AS TD
        LEFT JOIN (SELECT referenceID, count(referenceID) as Referred
        FROM paper_reference GROUP BY referenceID) AS TA
        ON TD.PaperID=TA.referenceID ORDER BY Referred DESC limit i, 10
```

7.6 Similar Papers

We choose top-10 most referred papers from the results of reference page and citation page. Use the UNION keyword to link two tables in the database.

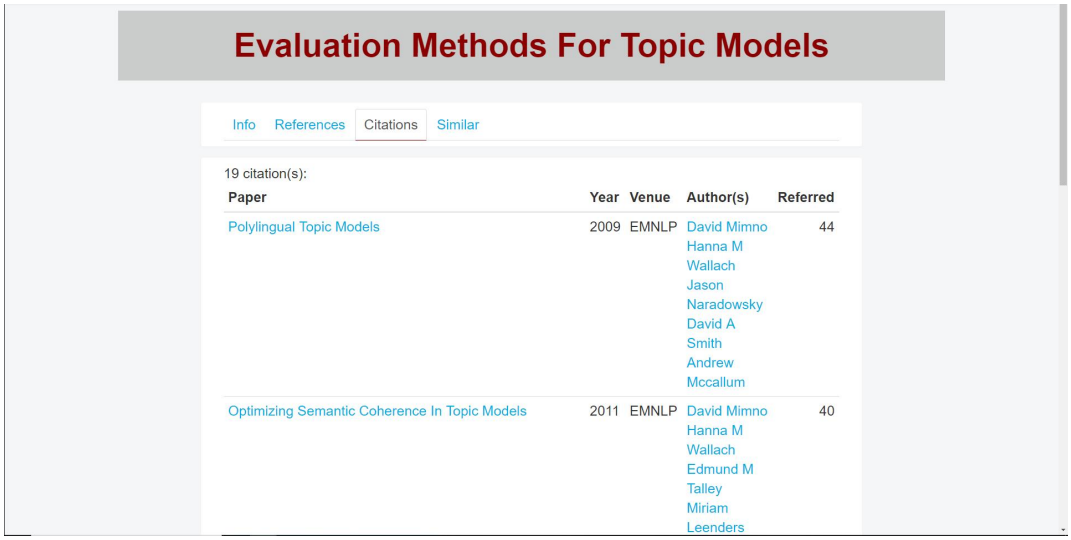


Figure 7.4: The paper's citation information.

```
SELECT Title, TD.PaperID, PaperPublishYear,
ConferenceID, Referred FROM
(SELECT TC.PaperID, Title, PaperPublishYear,ConferenceID
FROM (SELECT referenceID as PaperID FROM paper_reference
WHERE PaperID=\"{$_GET["PaperID"]}\"
UNION SELECT PaperID FROM paper_reference
WHERE referenceID=\"{$_GET["PaperID"]}\") AS TC
JOIN papers ON TC.PaperID=papers.PaperID) AS TD
LEFT JOIN (SELECT referenceID, count(referenceID) as Referred
FROM paper_reference GROUP BY referenceID) AS TA
ON TD.PaperID=TA.referenceID ORDER BY Referred DESC limit 0, 10
```

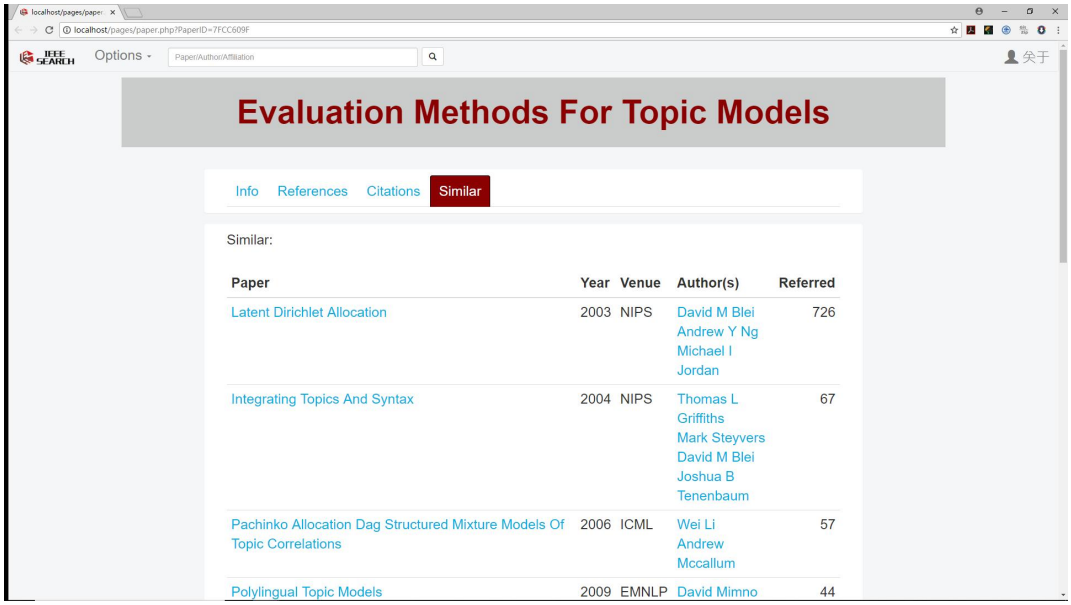


Figure 7.5: The similar paper information.

Chapter 8

Conference Page

8.1 Elements in Conference Page

Conference Page will show the information about this conference, such as:

- The line-chart of everyyear amount of the papers published in this conference.
- All the papers published in this conference.

8.2 Papers

The conference page will show all the papers in this conference. The top-10 most famous(referred mostly) papers will be shown on the first page. Use button to change the current page.

```
1         SELECT PaperID, Title, Year, Referred FROM
2         (SELECT PaperID, Title, PaperPublishYear AS Year FROM papers
3         WHERE ConferenceID = \"{$_GET["ConferenceID"]}\") AS TB
4         LEFT JOIN (SELECT referenceID, count(referenceID) as Referred
5         FROM paper_reference GROUP BY referenceID) AS TA
6         ON TB.PaperID=TA.referenceID ORDER BY Referred DESC limit i, 10
```

8.3 The Final Page

6242 paper(s) published in this venue:

Paper	Year	First Author	Referred
optimizing search engines using clickthrough data	2002	thorsten joachims	413
maximizing the spread of influence through a social network	2003	david kempe jon kleinberg eva tardos	249
mining and summarizing customer reviews	2004	minqing hu bing liu	208
the weka data mining software an update	2009	mark a hall eibe frank geoffrey holmes bernhard pfahringer peter reutemann ian h witten	194
training linear svms in linear time	2006	thorsten joachims	170

Figure 8.1: The conference paper information.

Figure 8.2: The conference page.

Chapter 9

Affiliation Page

9.1 Elements in Affiliation Page

Affiliation Page will show the information about this affiliation, such as:

- All the authors of this affiliation.
- All the papers published by the first author from this affiliation.
- Top-50 authors cooperation relationship graph.
- The line-chart of everyyear amount of papers from this affiliation.

9.2 Authors in the Affiliation

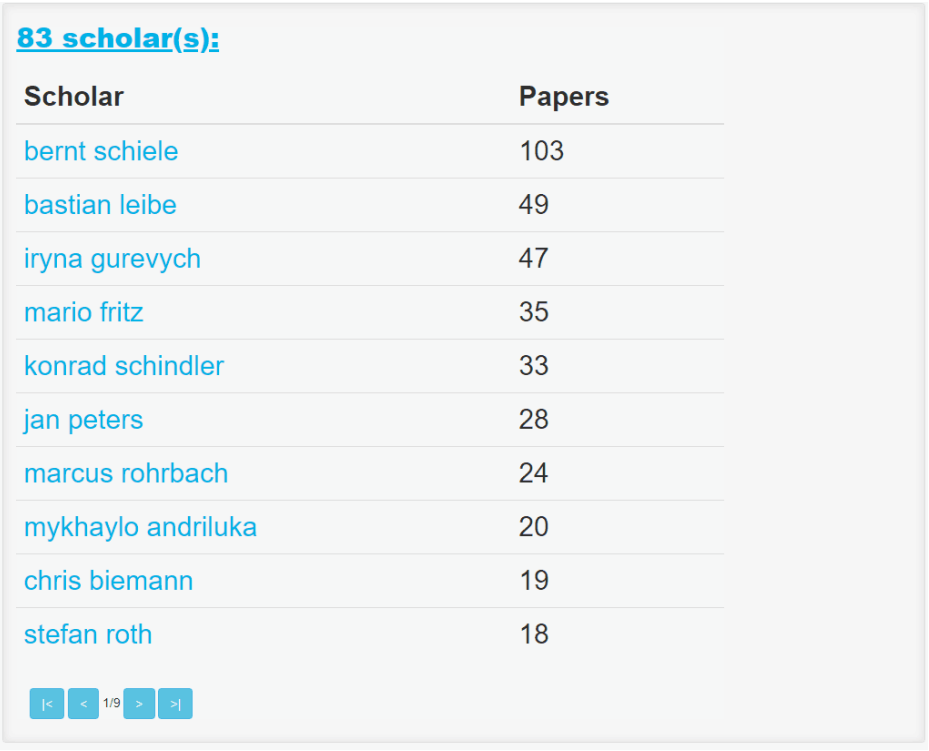
All the authors from this affiliation will be shown including the authors worked in the past. The would be sequenced by their papers amount.

```
1         SELECT TB.AuthorID, AuthorName, Papers FROM
2         (SELECT TA.AuthorID, AuthorName
3         FROM (SELECT AuthorID FROM paper_author_affiliation
4         WHERE AffiliationID=\"{$_GET["AffiliationID"]}\"
5         GROUP BY AuthorID) AS TA
6         LEFT JOIN authors ON TA.AuthorID=authors.AuthorID) AS TB
7         LEFT JOIN (SELECT AuthorID, count(*) AS Papers FROM
8         paper_author_affiliation GROUP BY AuthorID) AS TC
9         ON TB.AuthorID=TC.AuthorID ORDER BY Papers DESC limit i, 10
```

9.3 Papers published from this affiliation

All the papers published by the first author from this affiliation will be shown. Just like the Author Page, they will be sequenced by the referred times. Only the first author(i.e. AuthorSequence=1) will be shown in the paper's information.

```
1         SELECT TB.PaperID, PaperPublishYear, Title,
```



83 scholar(s):

Scholar	Papers
bernt schiele	103
bastian leibe	49
iryna gurevych	47
mario fritz	35
konrad schindler	33
jan peters	28
marcus rohrbach	24
mykhaylo andriluka	20
chris biemann	19
stefan roth	18

|< < 1/9 > >|

Figure 9.1: The scholars in this affiliation.

```
        ConferenceID, AuthorID, Referred FROM
        (SELECT papers.PaperID, PaperPublishYear,
        Title, ConferenceID, AuthorID FROM (
        SELECT PaperID, AuthorID FROM paper_author_affiliation
        WHERE AffiliationID = \"{$_GET['AffiliationID']}\"
        AND AuthorSequence = 1
        GROUP BY PaperID) AS TA
        LEFT JOIN papers ON TA.PaperID=papers.PaperID) AS TB
        LEFT JOIN (SELECT ReferenceID, count(ReferenceID) as Referred
        FROM paper_reference GROUP BY ReferenceID) AS TC
        ON TB.PaperID=TC.ReferenceID ORDER BY Referred DESC limit i, 10
```

9.4 Top-50 Authors Cooperation Relationship Graph

1. Use d3.js to show the graph. The fundamental array framework is as follows. We use one center point to represent the affiliation. It has a unique id and other dots will surrounding it.

```
                $authors = array();
                $allco1 = array(
                "nodes" => array(array("id" => "0",
                "name"=>$affiliationname,
                "group"=>0)),
                "links" => array()
```

50 paper(s) with the first authors in this affiliation:

Paper	Year	Venue	First Author	Referred
pedestrian detection in crowded scenes	2005	CVPR	bastian leibe	96
new features and insights for pedestrian detection	2010	CVPR	stefan walk	31
what helps where and why semantic relatedness for knowledge transfer	2010	CVPR	marcus rohrbach	31
a shape based object class model for knowledge transfer	2009	ICCV	michael m stark	25
discrete	2012	CVPR	anton	19

Figure 9.2: The papers published in this affiliation.

```
                );
```

2. Find the top-50 authors.

```
                SELECT TB.AuthorID, AuthorName, Papers FROM
                (SELECT TA.AuthorID, AuthorName
                FROM (SELECT AuthorID FROM paper_author_affiliation
                WHERE AffiliationID=\"{$_GET["AffiliationID"]}\"
                GROUP BY AuthorID) AS TA
                LEFT JOIN authors ON TA.AuthorID=authors.AuthorID) AS TB
                LEFT JOIN (SELECT AuthorID, count(*) AS Papers FROM
                paper_author_affiliation GROUP BY AuthorID) AS TC
                ON TB.AuthorID=TC.AuthorID ORDER BY Papers DESC limit 0, 50
```

3. The top-50 authors will be shown as dots. Their amount of papers will be the label of them. Use different colors to judge them.

```
while (($row=mysqli_fetch_array($result))) {
array_push($authors, $row['AuthorID']);
$allco1["nodes"][] = array(
"id" => $row['AuthorID'],
"name" => $row['AuthorName'],
//"group" => 1
"group" => intval(intval($row['Papers'])/25) + 1
);
}
```

4. Every dot has a invisible link to the affiliation dot(the value is 0).

```
for ($i = 0; $i < $aulen; ++$i) {
$allco1["links"][] = array(
"source" => "0",
"target" => $authors[$i],
"value" => 0
);
}
```

5. Find all cooperation just like the Author Page. Link the dots in accordance with the value of the cooperation times.

```
for ($i = 0; $i < $aulen; ++$i) {
$authorpaper[] = array();
$sql = "SELECT PaperID FROM paper_author_affiliation
WHERE AuthorID=\"{$authors[$i]}\"";
$result = mysqli_query($conn, "set names utf8");
$result = mysqli_query($conn, $sql);
while (($row=mysqli_fetch_array($result))) {
array_push($authorpaper[$i], $row['PaperID']);
}
}
for ($i = 0; $i < $aulen; ++$i) {
$allco1["links"][] = array(
"source" => "0",
"target" => $authors[$i],
"value" => 0
);
for ($j = $i + 1; $j < $aulen; ++$j) {
$conum = count(array_intersect($authorpaper[$i],
$authorpaper[$j]));
if ($conum != 0)
$allco1["links"][] = array(
"source" => $authors[$i],
"target" => $authors[$j],
"value" => $conum
);
}
}
```

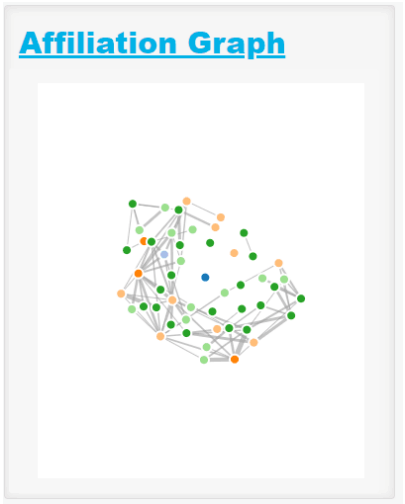


Figure 9.3: The top-50 authors cooperation relationship graph.

9.5 The final page

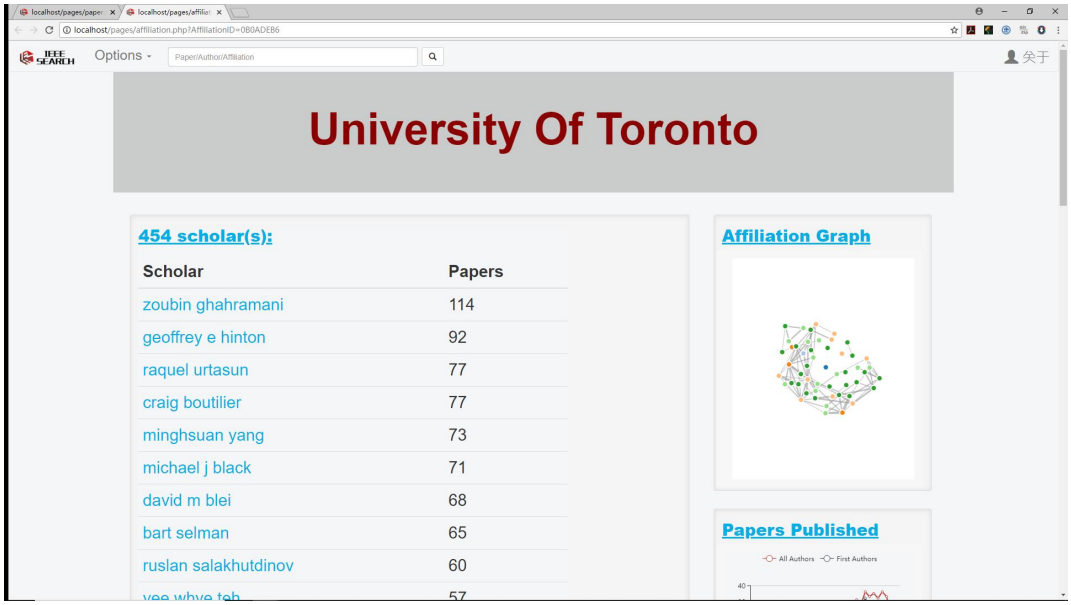


Figure 9.4: The affiliation page(1).

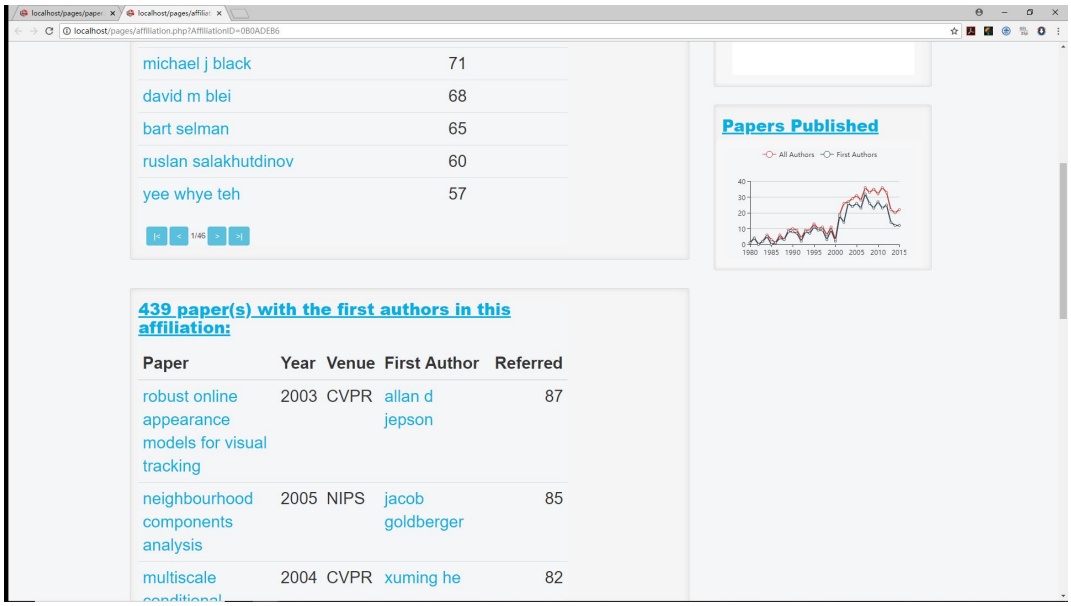


Figure 9.5: The affiliation page(2).

Line chart/Auxiliary page (QiuXuan Ling)

Chapter 10

Line chart

10.1 Paper-Author

10.1.1 Sourcecode

```
<script type="text/javascript" src="js/echarts.js"></script>

<script type="text/javascript">
var dom = document.getElementById("container");
var myChart = echarts.init(dom);

var app = {};
option = null;
option = {
tooltip: {
trigger: 'axis'
},
legend: {
data:['All Authors','First Authors']
},
grid: {
left: '3%',
right: '4%',
bottom: '3%',
containLabel: true
},
/*toolbox: {
feature: {
saveAsImage: {}
}
},*/
xAxis: {
type: 'category',
boundaryGap: false,
data: []
},
```

```
yAxis: {
type: 'value'
},
series: [
{
name:'All Authors',
type:'line',
//stack: '',
data:[]
},
{
name:'First Authors',
type:'line',
//stack: '',
data:[]
}
]
};
;
myChart.showLoading();
$.getJSON("affiliationpapers.php"+window.location.search, function(data1){
myChart.setOption({
xAxis: {
data: data1.years
},
series: [{
name: 'All Authors',
data: data1.all
},
{
name: 'First Authors',
data: data1.first
}]
});

});

if (option && typeof option === "object") {
myChart.hideLoading();
myChart.setOption(option, true);
}
</script>
```

10.1.2 Analysis

Introduction I used echart to make the line graphs. The first graph shows the the paper published each year by the author you have chosen.

And the blue line means the author is the first author of the published papers.

The graphs show the data before 2016, but we can see that, if the scholar stopped to publish papers before 2016, later data will be hidden.

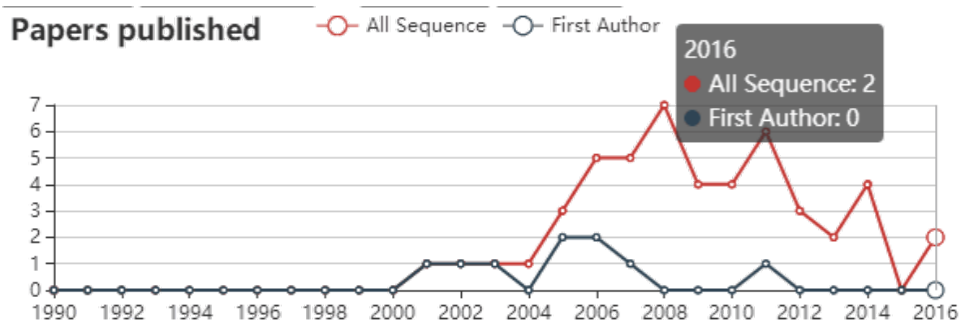


Figure 10.1: Line Chart 1

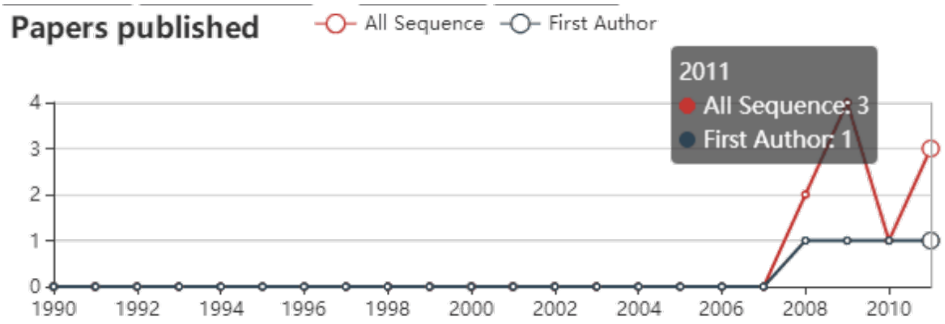


Figure 10.2: Line Chart 2

10.2 Paper-Affiliation

10.2.1 Sourcecode

```
<script type="text/javascript" src="js/echarts.js"></script>

<script type="text/javascript">
var dom = document.getElementById("container");
var myChart = echarts.init(dom);

var app = {};
option = null;
option = {
tooltip: {
trigger: 'axis'
},
legend: {
data:['All Authors','First Authors']
},
grid: {
left: '3%',
right: '4%',
```

```
bottom: '3%',
containLabel: true
},
/*toolbox: {
feature: {
saveAsImage: {}
}
},*/
xAxis: {
type: 'category',
boundaryGap: false,
data: []
},
yAxis: {
type: 'value'
},
series: [
{
name:'All Authors',
type:'line',
//stack: '',
data:[]
},
{
name:'First Authors',
type:'line',
//stack: '',
data:[]
}
]
};
;
myChart.showLoading();
$.getJSON("affiliationpapers.php"+window.location.search, function(data1){
myChart.setOption({
xAxis: {
data: data1.years
},
series: [{
name: 'All Authors',
data: data1.all
},
{
name: 'First Authors',
data: data1.first
}]
});

});

if (option && typeof option === "object") {
myChart.hideLoading();
```

```
myChart.setOption(option, true);
}
</script>
```

10.2.2 Analysis

Parts in affiliation.php

Introduction These line graphs show papers published each year from this affiliation. It's one of standard to evaluate the affiliation. .

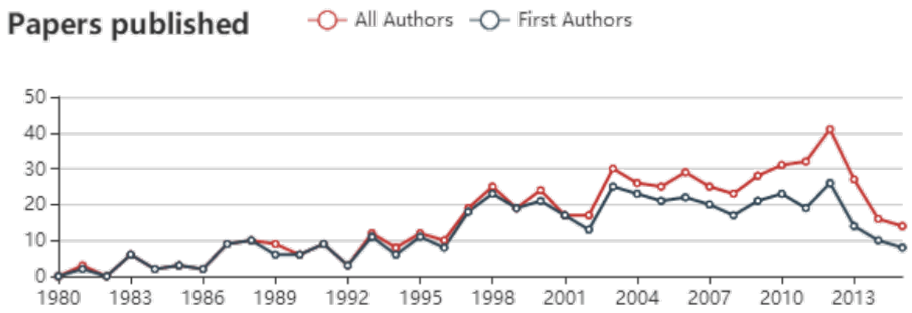


Figure 10.3: Line Chart 3

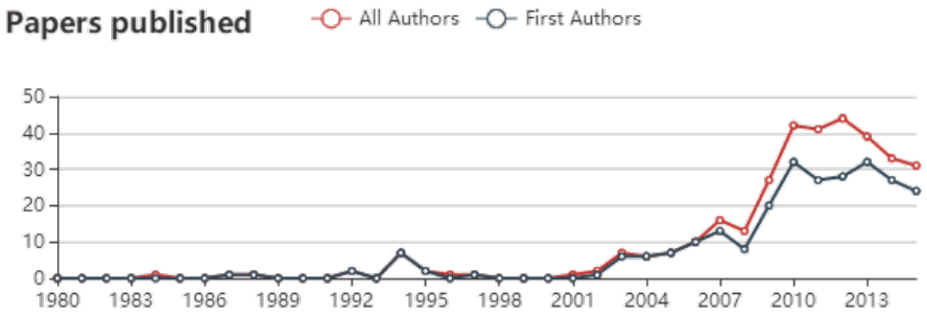


Figure 10.4: Line Chart 4

10.3 Paper-Venue

10.3.1 Sourcecode

```
<script type="text/javascript" src="js/echarts.js"></script>
<script type="text/javascript">
var dom = document.getElementById("container");
var myChart = echarts.init(dom);
```

```
5
6  var app = {};
7  option = null;
8  option = {
9  tooltip: {
10 trigger: 'axis'
11 },/*
12 legend: {
13 data:['Papers']
14 },*/
15 grid: {
16 left: '3%',
17 right: '4%',
18 bottom: '3%',
19 containLabel: true
20 },
21 /*toolbox: {
22 feature: {
23 saveAsImage: {}
24 }
25 },*/
26 xAxis: {
27 type: 'category',
28 boundaryGap: false,
29 data: []
30 },
31 yAxis: {
32 type: 'value'
33 },
34 series: [
35 {
36 name:'Papers',
37 type:'line',
38 //stack: □□'',
39 data:[]
40 }
41 ]
42 };
43 ;
44 myChart.showLoading();
45 $.getJSON("conferencepapers.php"+window.location.search, function(data1){
46 myChart.setOption({
47 xAxis: {
48 data: data1.years
49 },
50 series: [{
51 name: 'Papers',
52 data: data1.all
53 }]
54 });
55
56 });
```

```
57
58 if (option && typeof option === "object") {
59 myChart.hideLoading();
60 myChart.setOption(option, true);
61 }
62 </script>
```

10.3.2 Analysis

Introduction The third one shows the papers published in the venue you have chosen.

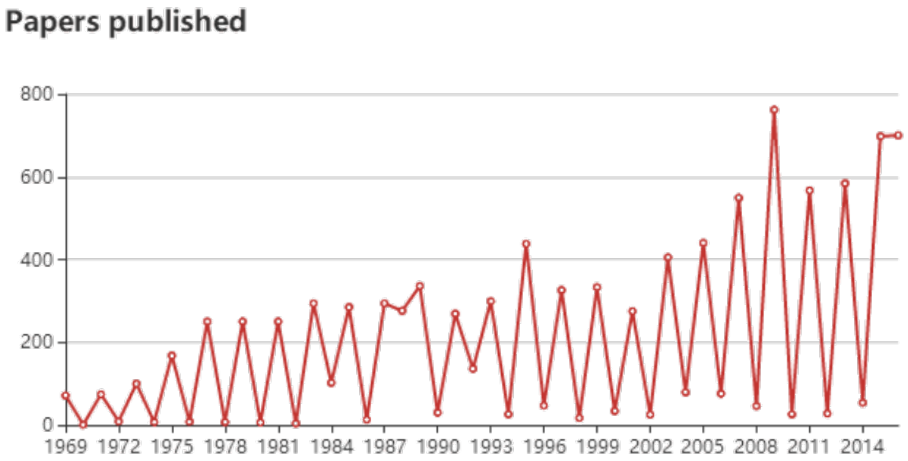


Figure 10.5: Line Chart 5

Chapter 11

Auxiliary page

11.1 Page: About Us

11.1.1 Sourcecode

About_Us.css

```
html,body,div,span,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,abbr,address,
    cite,code,del,dfn,em,img,ins,kbd,q,samp,small,strong,sub,sup,var,b,i,dl,dt,dd,ol
    ,ul,li,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,
    article,aside,canvas,details,figcaption,figure,footer,header,hgroup,menu,nav,
    section,summary,time,mark,audio,video {
border:0;
font:inherit;
font-size:100%;
margin:0;
padding:0;
vertical-align:baseline;
}

article,aside,details,figcaption,figure,footer,header,hgroup,menu,nav,section {
display:block;
}

html, body { font-family: 'Lato', helvetica, arial, sans-serif; font-size: 16px;
    color: #222;}

.clear {clear: both;}

p {
font-size: 1em;
line-height: 1.4em;
margin-bottom: 20px;
color: #444;
}

```

```
#cv {
width: 90%;
max-width: 800px;
background: #f3f3f3;
margin: 30px auto;
}

.mainDetails {
padding: 25px 35px;
border-bottom: 2px solid #cf8a05;
background: #ededed;
}

#name h1 {
color: grey;
font-size: 2.5em;
font-weight: 700;
font-family: 'Rokkitt', Helvetica, Arial, sans-serif;
margin-bottom: -6px;
}

#name h2 {
font-size: 1.5em;
margin-left: 2px;
font-family: 'Rokkitt', Helvetica, Arial, sans-serif;
}

#mainArea {
padding: 0 40px;
}
```

About_Us.html

```
<!DOCTYPE html>
<html>
<head>
<title>About Us</title>

<meta name="viewport" content="width=device-width"/>
<meta name="description" content="The Curriculum Vitae of Joe Bloggs."/>
<meta charset="UTF-8">

<link type="text/css" rel="stylesheet" href="style.css">
<link href='http://fonts.googleapis.com/css?family=Rokkitt:400,700|Lato:400,300'
    rel='stylesheet' type='text/css'>

<!--[if lt IE 9]>
<script src="//html5shiv.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
</head>
<body id="top">
```

```
<div id="cv" class="instaFade">
<div class="mainDetails">
<div id="headshot" class="quickFade">
<img src="headshot.jpg" alt="Alan Smith" />
</div>

<div id="name">
<h1 id="homeHeading">IEEE <span class="text-colored">Search</span></h1>
<h2 class="quickFade delayThree">Group 6</h2>
</div>


<div class="clear"></div>
</div>

<div id="mainArea" class="quickFade delayFive">
<section>
<article>
<div class="sectionTitle">
<h1>Website Introduction</h1>
</div>

<div class="sectionContent">
<p>This website is used for scholar search.<br>
The search bar on middle of the homepage can be retrieved by author/affiliation/
    conference/paper search.</p>
</div>
</article>
<div class="clear"></div>
</section>


<section>
<div class="sectionTitle">
<h1>Group Members</h1>
</div>

<div class="sectionContent">
<article>
<h2>Weidong Wang</h2>
<p class="subDetails">F1703014</p>
<p>E-mail address: wedonsjtu@sjtu.edu.cn</p>
</article>

<article>
<h2>Zhibang Wang</h2>
<p class="subDetails">F1703012</p>
<p>E-mail address: wswswzb@hotmail.com</p>
</article>

<article>
<h2>Xingzi Yu</h2>
```

```
<p class="subDetails">F1703012</p>
<p>E-mail address: edittriendl@sjtu.edu.cn</p>
</article>

<article>
<h2>Qiuxuan Ling</h2>
<p class="subDetails">F1703012</p>
<p>E-mail address: 2017lqx@sjtu.cn</p>
</article>
</div>
<div class="clear"></div>
</section>









</div>
</div>
<script type="text/javascript">
var gaJsHost = (("https:" == document.location.protocol) ? "https://ssl." : "http
    ://www.");
document.write(unescape("%3Cscript src='" + gaJsHost + "google-analytics.com/ga.js'
     type='text/javascript'%3E%3C/script%3E"));
</script>
<script type="text/javascript">
var pageTracker = _gat._getTracker("UA-3753241-1");
pageTracker._initData();
pageTracker._trackPageview();
</script>
<div class="ace-footer">
<p style="text-align: center">© IEEE 2018 Group 6</p>
</div>
</body>
</html>
```

11.1.2 Analysis

The codes are too long so we just chose a small part to present here to give an example.

Introduction In the page "About Us ", we have a brief introduction of our as well as the information of our group members.

Result Display .

CSS Framework We used the css framework to make the page we want. By adjusting some parameters, we finally got what we want.

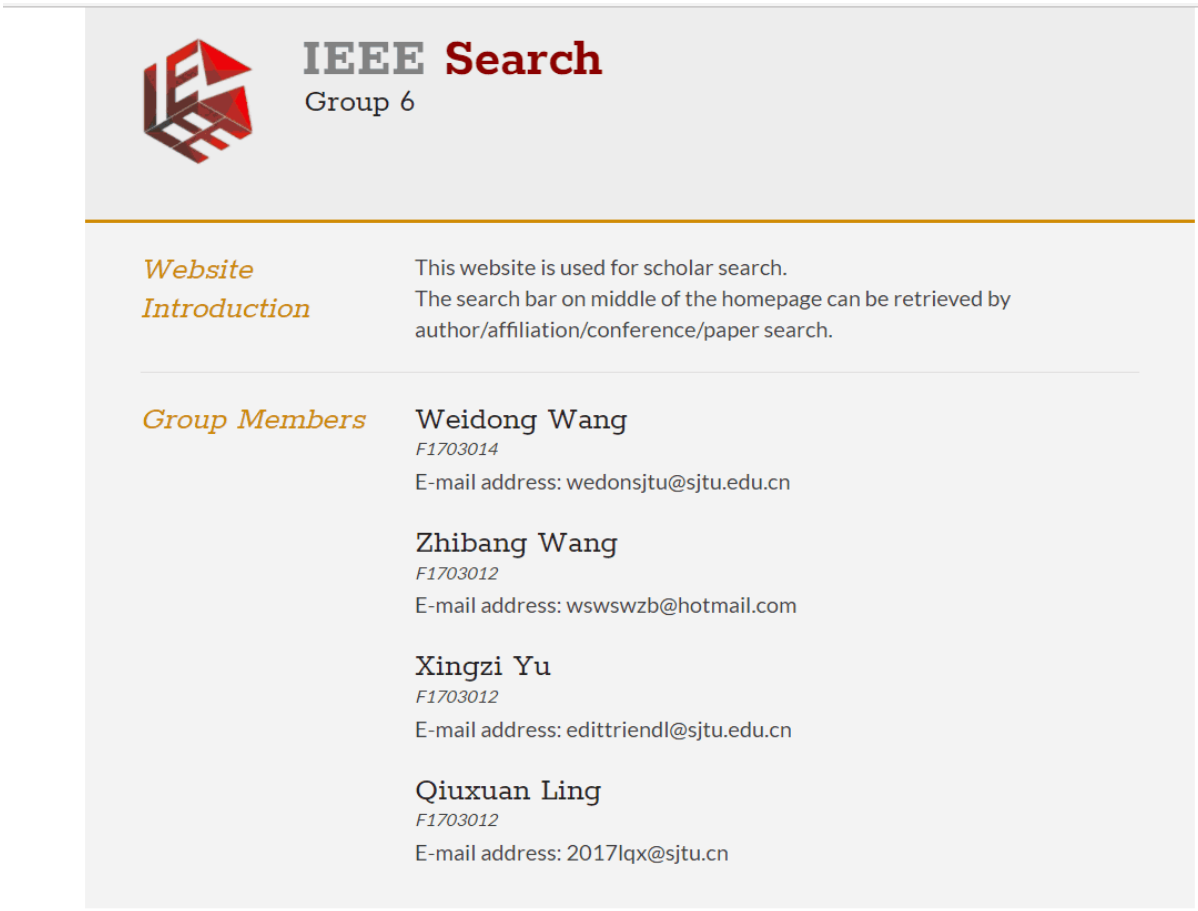


Figure 11.1: Page: About Us

html Then, we added the contents and linked it with other pages that we can turn to home page by clicking on the button.

11.2 Page: No Result

11.2.1 Sourcecode

No_Result.css

```
@-webkit-keyframes appear{
from{
opacity: 0;
}
to      {
opacity: 1;
}
}

@-webkit-keyframes headline_appear_animation{
from{
opacity: 0;
}
25% {
opacity: 0;
}
to      {
opacity: 1;
}
}

@-webkit-keyframes contentappear{
from {
-webkit-transform: scale(0);
opacity: 0;
}
50% {
-webkit-transform:  scale(0.5);
opacity: 0;
}
to {
-webkit-transform: scale(1);
opacity: 1;
}
}

@-moz-keyframes appear{
from{
opacity: 0;
}
to      {
```

```
opacity: 1;
}
}

@-moz-keyframes headline_appear_animation{
from{
opacity: 0;
}
25% {
opacity: 0;
}
to      {
opacity: 1;
}
}

@-moz-keyframes contentappear{
from {
-moz-transform: scale(0);
opacity: 0;
}
50% {
-moz-transform:  scale(0.5);
opacity: 0;
}
to {
-moz-transform: scale(1);
opacity: 1;
}
}

* {
margin: 0;
padding: 0;
}
```

No_Result.html

```
<!doctype html>
<html lang="en">

<header>
<meta charset="utf-8">
<title>No Such Result</title>

<link rel="stylesheet" media="screen" href="404.css">

<script language="JavaScript">
function dosearch() {
var sf=document.searchform;
var submitto = sf.sengine.value + escape(sf.searchterms.value);
```

```
window.location.href = submitto;
return false;
}
</script>
<div class="header-content">
<div class="header-content-inner">
<h1 id="homeHeading">IEEE <span class="text-colored">Search</span></h1>
<hr>
</div>
</div>
</header>


<body>

<div class="content">
<h2>No Such Result</h2>

<p class="text">…
Oooooops <br>
It looks like the author/affiliation/conference/paper you were looking for does not exist.

<form name="searchform" onSubmit="return dosearch();">
<input type="hidden" name="sengine" value="http://www.google.com/search?q=site:www.yoursite.com+" />
<input type="text" name="searchterms" class="inputform">
<input type="submit" name="SearchSubmit" value="Search" class="button">
</form>
<!-- Change www.yoursite.com to your website domain -->
</p>

<p class="links">
<a href="#">Homepage</a>  <a href="#">About Us</a>
<!--These are links. You can change it to a page you want to by replacing the '#' with your url.-->
</p>
</div>

</html>
```

11.2.2 Analysis

The codes are too long so we just chose a small part to present here to give an example.

Introduction If there is no result for what you want to search, you will jump to the page "No Result". Click the home bottom and you can get back to the home page.

Result Display .

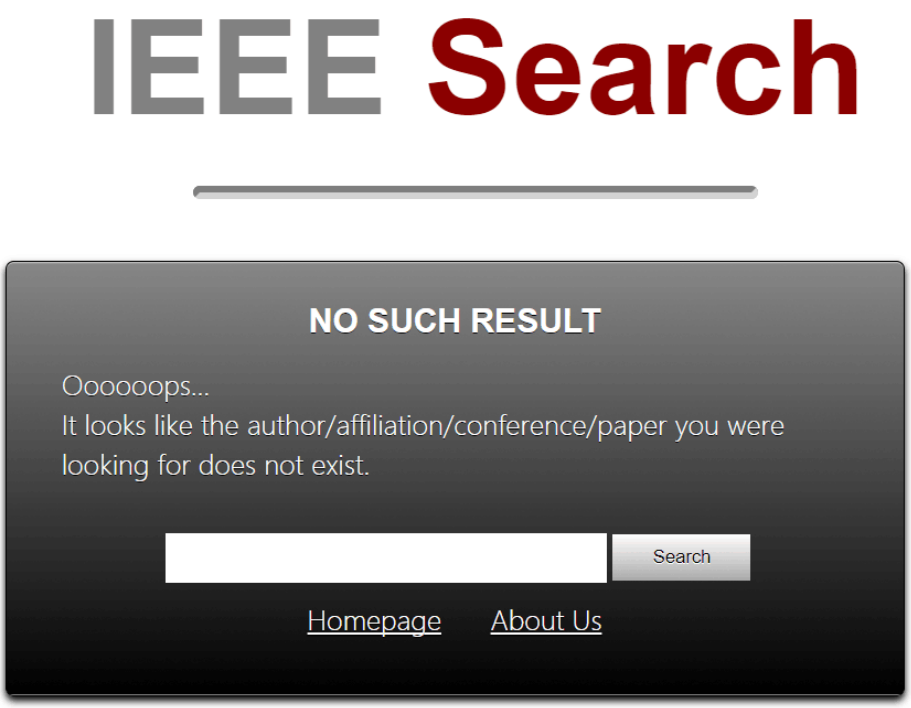


Figure 11.2: Page: No Result

CSS Framework We used the css framework to make the page we want. By adjusting some parameters, we finally got what we want just as the same as what we did before.

Html Then, we added the contents and linked it with other pages that we can turn to home or start to search at this page by clicking on the button or using the search box.