Final Project Report

Li Jingyu, He Wendi, Zhang Yuheng

June 12, 2018

Contents

Abstract

With the increasing academic papers included in online databases, websites that facilitate the search of papers are gaining popularity. In this project, we developed a website doing academic search. More than that, it contained other features that are user-friendly.

1 Concepts and Methods

1.1 MVC pattern

Introduction to MVC

MVC, is the acronym of Model–view–controller. It is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts, Model, View and Controller. This is done to separate internal representations of information from the ways information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development.

Components of MVC

- The model is the central component of the pattern. It expresses the application's behavior in terms of the problem domain, independent of the user interface. It directly manages the data, logic and rules of the application.
- A view can be any output representation of information, such as a chart or a diagram. Multiple views of the same information are possible, such as a bar chart for management and a tabular view for accountants.
- The third part or section, the controller, accepts input and converts it to commands for the model or view.

Interaction of MVC

The model is responsible for managing the data of the application. It receives user input from the controller.

The view means presentation of the model in a particular format.

The controller responds to the user input and performs interactions on the data model objects. The controller receives the input, optionally validates it and then passes the input to the model.

CodeIgniter

CodeIgniter is an open-source software rapid development web framework, for use in building dynamic web sites with PHP. It is loosely based on the popular model–view–controller (MVC) development pattern. While controller classes are a necessary part of development under CodeIgniter, models and views are optional. CodeIgniter is most often noted for its speed when compared to other PHP frameworks. It's a suitable tool for the code transplantation of this project.

1.2 CSS

Brief introduction to CSS

CSS refers to the Cascading Style Sheets. The style defines how to display the HTML element. Styles are usually stored in style sheets.Adding styles to HTML 4 is to solve the problem of separation of content from presentation.External style sheets can greatly improve work efficiency and are usually stored in CSS files. Multiple style definitions can be stacked into one.

The basic grammar of CSS

1. The CSS rule consists of two main parts: selector and one or more declarations.Just like: selector declaration1; declaration2; ... declarationN .

2. Selectors are usually HTML elements that you need to change styles. Each statement is made up of one attribute and one value. The attribute is the style property that you want to set. Each attribute has a value. Properties and values are separated by a colon. The writing method: selector property: value.
3. In order to understand better,we set an example. The role of the following code is to define the text color in the H1 element as red and set the font size to 14 pixels.

```
h1 {color:red; font-size:14px;}
```

In this example, H1 is a selector, color and font-size are attributes, red and 14px are values.

The selector of CSS

1. Descendent Selector.

   This kind of selector define the style by the context of the element in its location. For example, if you want the strong element in the list to be italic instead of the usual bold font, you can define a derived selector.
2. ID Selector.

   The ID selector can specify a specific style for the HTML element that has a specific ID.It is defined by ”#”. The following two ID selectors, the first one can define the color of the element is red, and the second elements are green.

```
#red {color:red;}
#green {color:green;}
<p id=”red”>This is red</p>
<p id=”green”>This is green</p>
```

   In modern layout, ID selectors are often used to build derivative selectors.
3. Class Selector.

   In CSS, the class selector is displayed on a point number:

```
.center {text-align: center}
```

   In the above example, all HTML elements with center class are centered. In the following HTML code, both H1 and P elements have center classes. This means that both will comply with the rules in the ”.Center” selector.

```
<h1 class=”center”>
This heading will be center-aligned
</h1>
<p class=”center”>
This paragraph will also be center-aligned.
</p>
```

   Like ID, class can also be used as a derived selector:

```
.fancy td {
        color: #f60;
        background: #666;
        }
```

   Elements can also be selected based on their classes:

```
td.fancy {
color: #f60;
          background: #666;
         }
```

4. Attribute Selectors.

   You can set styles for HTML elements with specified properties, not just class and ID attributes. The following example sets styles for all elements with title properties.

```
[title]
{
color:red;
}
<h2 title=”Hello world”>Hello world</h2>
```

Some introduction to Bootstrap

Bootstrap is a front-end framework for rapidly developing Web applications and web sites. Bootstrap is based on HTML, CSS, and JAVASCRIPT.

1. Basic structure: Bootstrap provides a basic structure with grid system, link style and background. This will be explained in detail in the basic structure part of Bootstrap.
2. CSS:Bootstrap has the following features: the global CSS setting, the definition of the basic HTML element style, the extensible class, and an advanced grid system. This will be explained in detail in the Bootstrap CSS section.
3. Component: Bootstrap contains more than a dozen reusable components for creating images, drop-down menus, navigation, warning boxes, pop-up boxes, and so on. This will be explained in detail in the layout component part.
4. JavaScript plug-in: Bootstrap contains more than a dozen custom jQuery plug-ins. You can directly include all plug-ins, or you can include these plug-ins one by one. This will be explained in detail in the Bootstrap plug-in section.
5. Customization: you can customize Bootstrap components, LESS variables and jQuery plug-ins to get your own version.

1.3 jquery

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

jQuery, at its core, is a Document Object Model (DOM) manipulation library. The DOM is a tree-structure representation of all the elements of a Web page. jQuery simplifies the syntax for finding, selecting, and manipulating these DOM elements. For example, jQuery can be used for finding an element in the document with a certain property (e.g. all elements with an h1 tag), changing one or more of its attributes (e.g. color, visibility), or making it respond to an event (e.g. a mouse click).

jQuery also provides a paradigm for event handling that goes beyond basic DOM element selection and manipulation. The event assignment and the event callback function definition are done in a single step in a single location in the code. jQuery also aims to incorporate other highly used JavaScript functionality (e.g. fade ins and fade outs when hiding elements, animations by manipulating CSS properties).

The principles of developing with jQuery are:

- Separation of JavaScript and HTML: The jQuery library provides simple syntax for adding event handlers to the DOM using JavaScript, rather than adding HTML event attributes to call JavaScript functions. Thus, it encourages developers to completely separate JavaScript code from HTML markup.
- Brevity and clarity: jQuery promotes brevity and clarity with features like chainable functions and shorthand function names.
- Elimination of cross-browser incompatibilities: The JavaScript engines of different browsers differ slightly so JavaScript code that works for one browser may not work for another. Like other JavaScript toolkits, jQuery handles all these cross-browser inconsistencies and provides a consistent interface that works across different browsers.
- Extensibility: New events, elements, and methods can be easily added and then reused as a plugin.

1.4 d3

Introduction to D3

D3.js (or just D3 for Data-Driven Documents) is a JavaScript library for producing dynamic, interactive data visualizations in web browsers. It makes use of the widely implemented SVG, HTML5, and CSS standards. It is the successor to the earlier Protovis framework. In contrast to many other libraries, D3.js allows great control over the final visual result. Its development was noted in 2011, as version 2.0.0 was released in August 2011.

D3.js is used on hundreds of thousands of websites. Some popular uses include creating interactive graphics for online news websites, information dashboards for viewing data, and producing maps from GIS map making data. In addition, the exportable nature of SVG enables graphics created by D3 to be used in print publications.

Technical Principles

- Selections

  The central principle of D3.js design is to enable the programmer to first use a CSS-style selector to select a given set of Document Object Model (DOM) nodes, then use operators to manipulate them in a similar manner to jQuery.[9] For example, by using D3.js, one may select all HTML <p>...</p> elements, and then change their text color, e.g. to lavender:
- Transitions

  By declaring a transition, values for attributes and styles can be smoothly interpolated over a certain time. The following code will make all HTML <p>...</p> elements on a page gradually change their text color to pink:
- Data-binding

  For more advanced uses, loaded data drives the creation of elements. D3.js loads a given dataset, then, for each of its elements, creates an SVG object with associated properties (shape, colors, values) and behaviors (transitions, events)
- Appending nodes using data

  Once a dataset is bound to a document, use of D3.js typically follows a pattern wherein an explicit .enter() function, an implicit ”update,” and an explicit .exit() function is invoked for each item in the bound dataset. Any methods chained after the .enter() command will be called for each item in the dataset not already represented by a DOM node in the selection (the previous selectAll()). Likewise, the implicit update function is called on all existing selected nodes for which there is a corresponding item in the dataset, and .exit() is called on all existing selected nodes that do not have an item in the dataset to bind to them. The D3.js documentation provides several examples of how this works.

2 Basic Function

We use the Model-View-Controller (MVC) developement pattern. Here is the layout of all of our component php files.

Model

- *ace_model.php*

Controller

- *acemap.php*

View

- *header.php*
- *footer.php*
- *home.php*
- *menu.php*
- *result_page.php*
- *author.php*
- *graph.php*
- *recommendation.php*
- *result_paper_page.php*
- *paper.php*
- *result_conf_page.php*
- *conf.php*

2.1 Search

To search information, the users can directly input an author name, or a paper name, or a conference name in the searching window and then click the optional button of “Name”,” Paper” or “Conference” according to what he has just input. After pressing the “submit” bottom, the home page will send the querying content and the type of the content to the result page.

The following code shows how the optional bottom be realized. We can see that the three choices have three different values and these values stand for their identities. Besides, it is a single optional bottom and “checked” means it will automatically choose the “Name” bottom if the user does not change it.

```
<form action=”” method=”post” id=”myform”>
    <input type=”input” id=”tags” name=”authorname” style=”
        color: black;” />
    <input type=”submit” name=”submit” value=”Search” id=”
        submit_btn” class=”btn btn-success”/>
    <br><br>
    <input type=”radio” name=”check” id=”au” checked>Author
        <input type=”radio” name=”check” id=”pa”>Paper
        <input type=”radio” name=”check” id=”co”>Conference
</form>
```

To submit form to different controller functions based on the user’s choice, we use the javascript to get the status of the checkbox and change the target page. You can see that they call different controller function to load the author page, paper page and conference page correspondingly.

```
$(”#au”).click(function(){
        $(”#info”).text(”Enter the scholar’s name”);
});
$(”#pa”).click(function(){
        $(”#info”).text(”Enter the paper’s title”);
});
$(”#co”).click(function(){
        $(”#info”).text(”Enter the conference’s name”);
});
$(”#submit_btn”).bind(”click”,function(){
        if($(”#au”).is(”:checked”)){
                $(”#myform”).attr(”action”,”result_page”);
                $(”#myform”).submit();
        }
        if($(”#pa”).is(”:checked”)){
                $(”#myform”).attr(”action”,”result_paper_page
                    ”);
                $(”#myform”).submit();
        }
        if($(”#co”).is(”:checked”)){
                $(”#myform”).attr(”action”,”result_conf”);
                $(”#myform”).submit();
        }
});
```

In addition, with the help of javascript code, we can transfer what the users are entering in real time to the hint page where it can query information from the database and return the related author’s name with the form of json.

So, when the user is entering in the querying window, it can automatically turn out a set of content which are closely related to what the user is just entering. So, even though the author has’t finish enter the full name yet, what the user probably wants to input will already be showed on the screen in a set of content listed.

```
$(function () {
        if($(”#au”).is(”:checked”)){
                $(”#tags”).autocomplete({
                source: ”autocomp”,
                minLength: 3
        });
        }
});
```

To get the desired data in the form of json, the following function is in the model.

```
public function get_auto($authorname)
{
        $query=$this->db->query("select author.name from
            author,relation where author.name like '
            $authorname' and relation.authorid=author.id
            group by author.name order by count(*) desc limit
             10");
        $counts=0;
        foreach($query->result_array() as $row){
                $result[$counts]=$row['name'];
                $counts++;
        }
        echo json_encode($result);
}
```

Here is our home page.

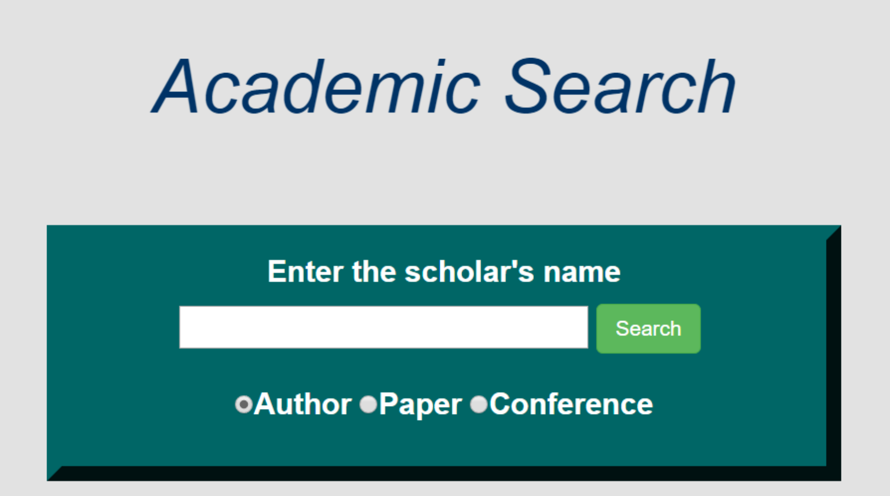
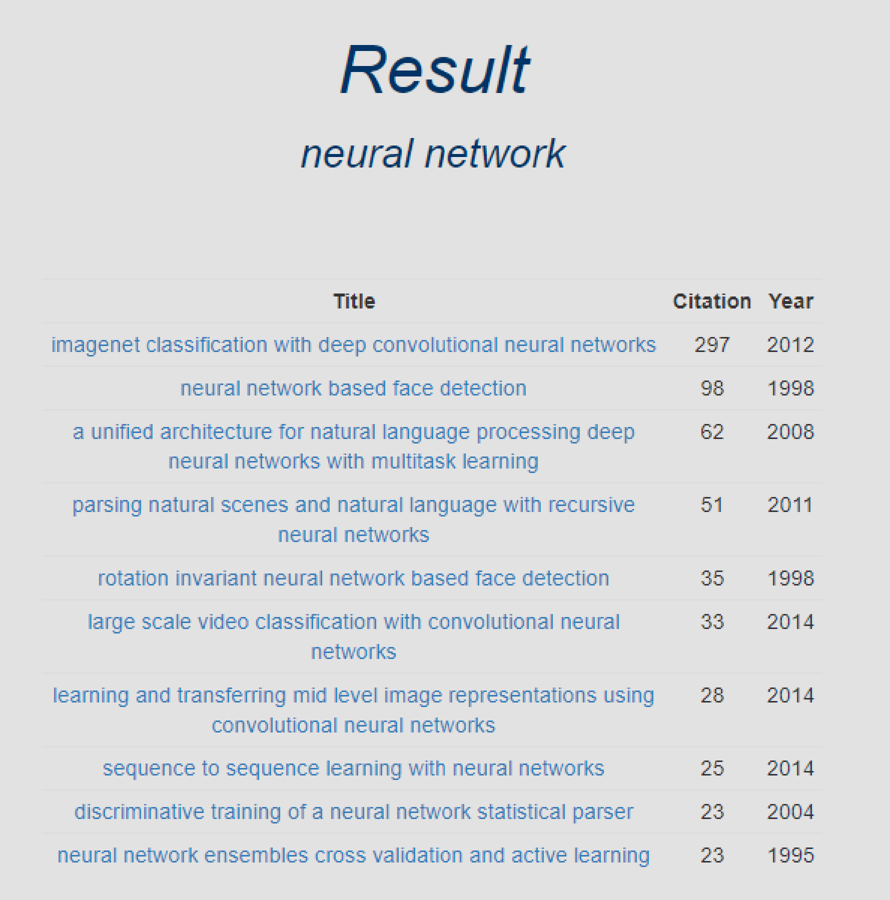


Figure 1: The homepage.

2.2 Page

2.2.1 Result Page



Result

neural network

Title	Citation	Year
imagenet classification with deep convolutional neural networks	297	2012
neural network based face detection	98	1998
a unified architecture for natural language processing deep neural networks with multitask learning	62	2008
parsing natural scenes and natural language with recursive neural networks	51	2011
rotation invariant neural network based face detection	35	1998
large scale video classification with convolutional neural networks	33	2014
learning and transferring mid level image representations using convolutional neural networks	28	2014
sequence to sequence learning with neural networks	25	2014
discriminative training of a neural network statistical parser	23	2004
neural network ensembles cross validation and active learning	23	1995

Figure 2: The result page of paper.

In result page, the querying content received from the searching page will be dealt with correspondingly and be reference to different querying sentence according to the different values of “Name”, “Paper” or “Conference”.

By using MySQL language and use “fuzzy search” in the page, it will query from the database and list all the relating information which are close to the content the user input. So, it means the user may not input the exact name of what he is looking for, but just a similar one instead and it will automatically show the candidates content.

Besides, the content showed in the result page are added with hyperlink so that the user can easily click the content he likes to look for further detail information of the content.

The following code take the choice of ”paper” as an example.

In the model, get some info about paper.

```
public function get_some_title($title ,$page)
{
        $begin=10*($page-1);
        $query=$this->db->query(”select title,pubyear,id,citation from paper left join (select refid,count(*) as citation from reference group by refid) t1 on t1.refid=paper.id where title like ’$title’ order by citation desc limit $begin,10”);
        return $query->result_array();
}
```

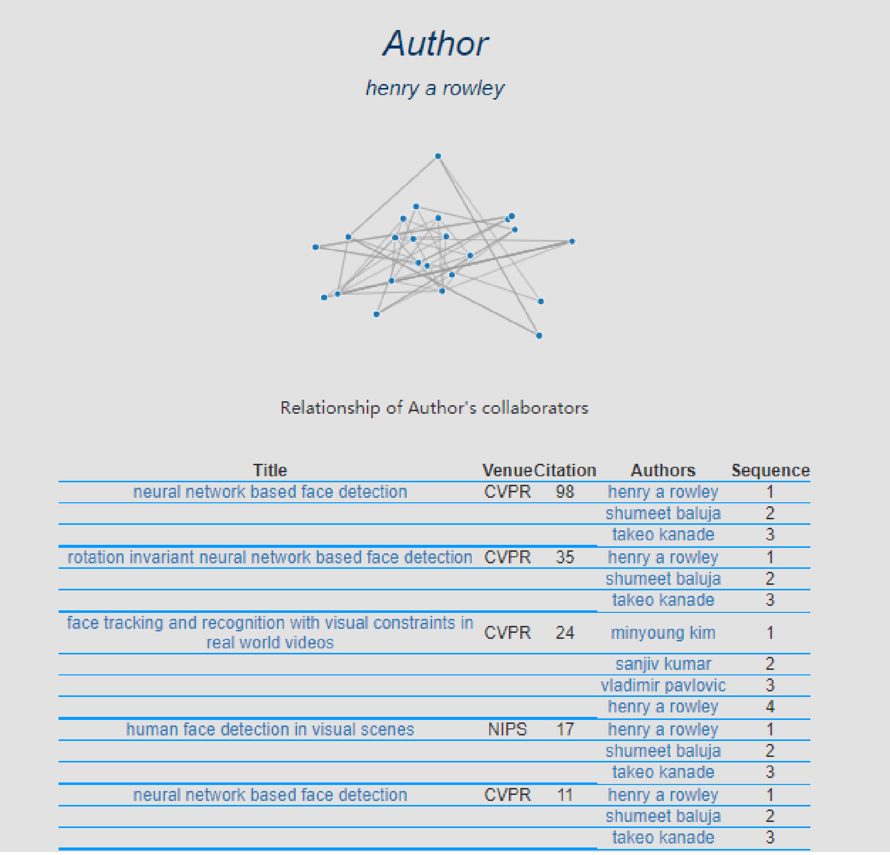
Note that we include pagination, but we will have a detailed discussion later. The view page is showned below. It mainly contains some tables and buttons.

```
<table class=”table table-condensed”>
        <th>Title</th>
        <th>Citation</th>
        <th>Year</th>
<?php foreach ($items as $item): ?>
        <tr>
        <td><a href=’paper_page/<?=$item[’id’]?>’><?=$item[’title’]?></a></td>
        <td style=’text-align: center’><?php if($item[’citation’]){echo $item[’citation’];}else{echo ”0”;}?></td>
        <td style=’text-align: center’><?=$item[’pubyear’]?></td>
        </tr>
<?php endforeach; ?>
</table>
```

The controller works as a bridge between the model and the view page.

```
public function result_paper_page()
{
    $title=$this->input->post(”authorname”);
    $page=1;
    $data[’title’]=”Result”;
    $data[’subtitle’]=$title;
    $data[’page’]=$page;
    $data[’items’]=$this->ace_model->get_some_title($title,$page);
    $data[’total’]=$this->ace_model->get_total_title($title);

    $this->load->view(’acemap/header’,$data);
    $this->load->view(’acemap/menu’);
    $this->load->view(’acemap/result_paper_page’,$data);
    $this->load->view(’acemap/footer’);
}
```

2.2.2 Author Page



Author

henry a rowley

Relationship of Author's collaborators

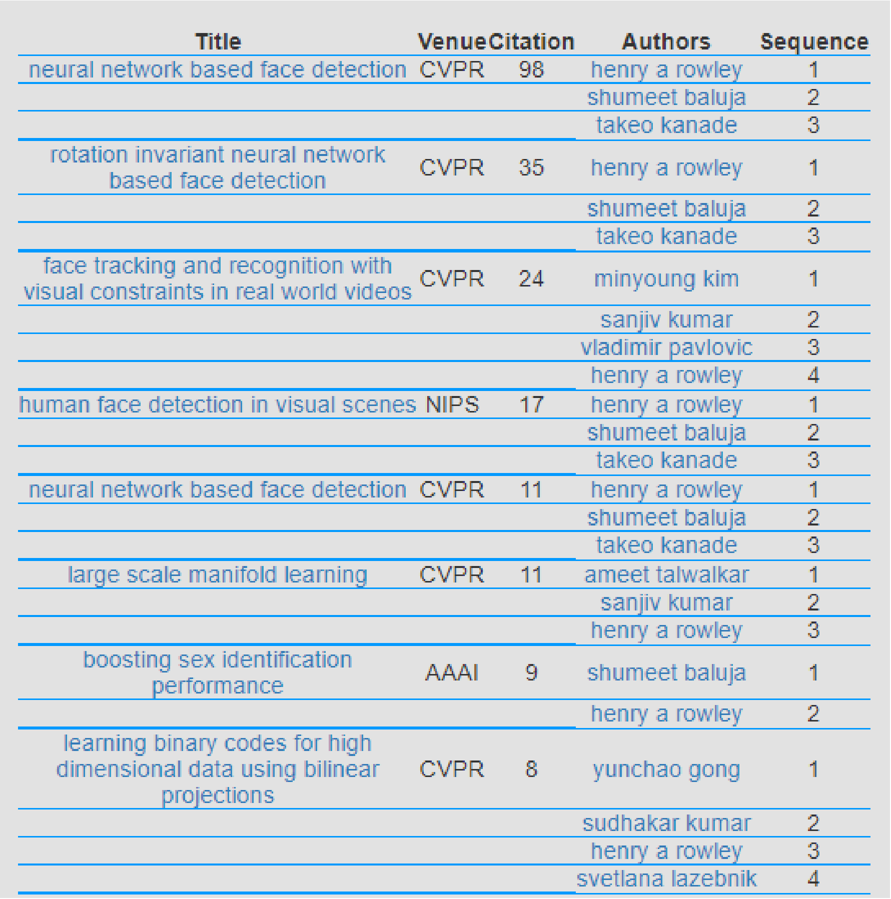
Title	Venue	Citation	Authors	Sequence
neural network based face detection	CVPR	98	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
rotation invariant neural network based face detection	CVPR	35	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
face tracking and recognition with visual constraints in real world videos	CVPR	24	minyoung kim	1
			sanjiv kumar	2
			vladimir pavlovic	3
			henry a rowley	4
human face detection in visual scenes	NIPS	17	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
neural network based face detection	CVPR	11	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3

Figure 3: The author page.

In the author page, by using a series of querying sentence and with the help of all kind of information in the database, we can see all the papers of the author ordered by the numbers of the reference paper. Besides, the conference of the author and the collaborate authors ordered by the author sequence are also showed.

In addition, the information showed in this page including the different paper written by the author, conference of the paper, the collaborate authors are added with hyperlinks so you can easily click them to look for further detail information of the paper, the conference and the authors. In a word, these pages are being closely related and can be referred to each other which make the website more convenient and helpful.

```
<table>
<th>Title</th>
<th>Venue</th>
<th>Citation</th>
<th>Authors</th>
<th>Sequence</th>
<?php foreach ($items as $item): ?>
        <tr>
        <td><a href=”/CI/index.php/acemap/paper_page/<?=$item[’id’]?>”><?=$item[’title’]?></a></td>
        <td><?=$item[’venue’]?></td>
        <td><?php if($item[’citation’])echo $item[’citation’];else echo ’0’?></td>
<?php foreach ($item[’authors’] as $authorinfo): ?>
<?php if(count($item[’authors’])>1){
                echo ”<td nowrap=’nowrap’><a href=’”.$authorinfo[’id’].”’>”.$authorinfo[’name’].”</a></td>”;
                echo ”<td>”.$authorinfo[’authorseq’].”</td>”;
                echo ”</tr>”;
                echo ”<tr><td></td><td></td><td></td>”;
        } else {
                echo ”<td>”.$authorinfo[’name’].”</td>”;
                echo ”<td>”.$authorinfo[’authorseq’].”</td>”;
                echo ”</tr>”;
        }
?>
<?php endforeach; ?>
        </tr>
<?php endforeach; ?>
</table>
```



Title	Venue	Citation	Authors	Sequence
neural network based face detection	CVPR	98	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
rotation invariant neural network based face detection	CVPR	35	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
face tracking and recognition with visual constraints in real world videos	CVPR	24	minyoung kim	1
			sanjiv kumar	2
			vladimir pavlovic	3
			henry a rowley	4
human face detection in visual scenes	NIPS	17	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
neural network based face detection	CVPR	11	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
large scale manifold learning	CVPR	11	ameet talwalkar	1
			sanjiv kumar	2
			henry a rowley	3
boosting sex identification performance	AAAI	9	shumeet baluja	1
			henry a rowley	2
learning binary codes for high dimensional data using bilinear projections	CVPR	8	yunchao gong	1
			sudhakar kumar	2
			henry a rowley	3
			svetlana lazebnik	4

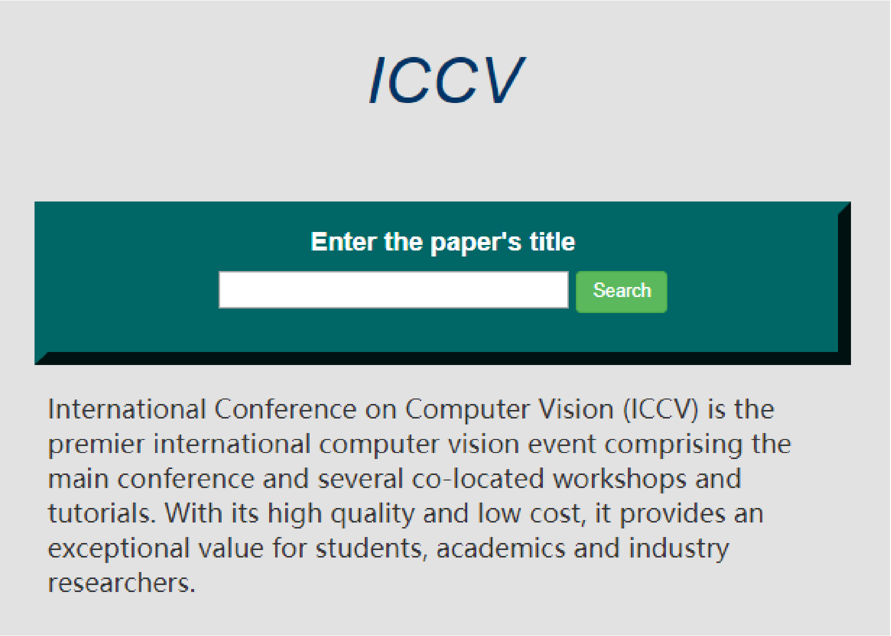
Figure 4: The table in the author page.

We also show the relationship of all the collaborators of the author in the form of relation graph. We will talk about it in the UI design part.

2.2.3 Paper Page

In the paper page, the user can see the name of the paper and the publish year of the paper. Also, all the authors of the paper are listed in the order of the author sequence. In addition, the names of authors are added with hyperlink. So, it will turn to the author page of that author if the user click that author's name to look for further information of that author. It's quite like the author page.

2.2.4 Conference Page



International Conference on Computer Vision (ICCV) is the premier international computer vision event comprising the main conference and several co-located workshops and tutorials. With its high quality and low cost, it provides an exceptional value for students, academics and industry researchers.

Figure 5: The conference page.

In the conference page, the user can see all the papers published on that conference. As there are so many papers, the publish year and the authors of the paper are also showed to let the user have a brief impression of the paper published on that conference. Besides, a brief introduction of the conference is added on each conference page.

Above the brief introduction, we added another search box to search the paper published in this conference.

We combine three different pages together by hyperlinks which compose a huge net so the user can look for information that they are interested in accessibly and conveniently.

2.3 Pagination

If one page contains too many information, it causes a sense of chaos and confusion. So we need pagination.

We also take the paper result page as an example.

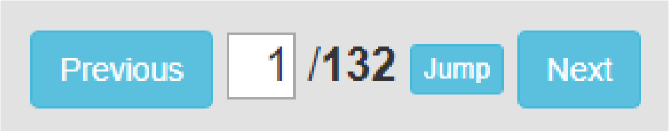


Figure 6: Page turning buttons.

First, we added some buttons related to pagination.

```
<button id=’btnPre’ class=”btn btn-info”>Previous</button>
```

```
<input type=”text” style=”width:30px;text-align:right” name=”
    page” value=1>
        /<label><?=$total?></label>
<button id='btnJump' class=”btn btn-info btn-xs”>Jump</button
    >
<button id='btnNext' class=”btn btn-info”>Next</button>
```

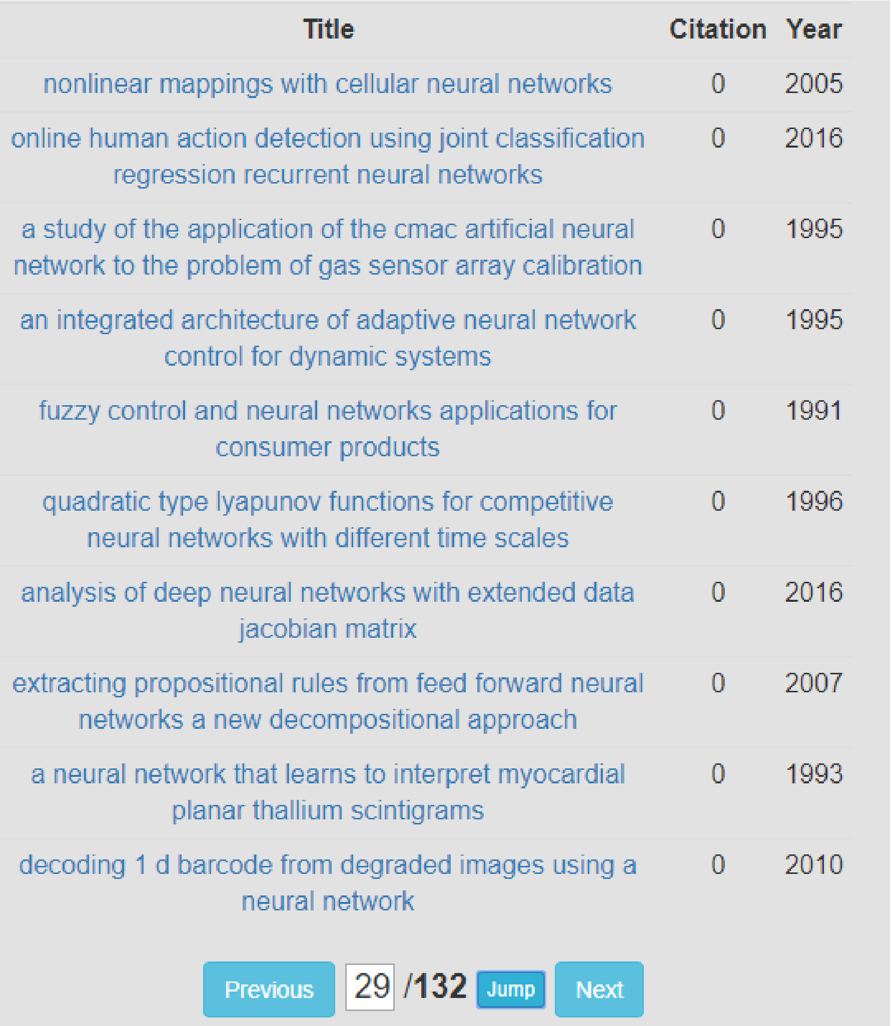
Then we use javascript to manipulate these buttons. We use the next page to turn to the next page, and previous for the opposite. Users can also input the page they want to browse and jump to that page.

```
var title=$(”h2”).text();
var page=parseInt($(”input”).val());
var item;
$(”#btnPre”).click(function(){
        page-=1;
        $.post(”/CI/index.php/acemap/new_result_paper_page
            ”,{”title”:title,”page”:page},function(data,
            status){
                $(”input”).val(page);
                $(”table”).replaceWith(data);
        })
});
$(”#btnNext”).click(function(){
        page+=1;
        $.post(”/CI/index.php/acemap/new_result_paper_page
            ”,{”title”:title,”page”:page},function(data,
            status){
                $(”input”).val(page);
                $(”table”).replaceWith(data);

        })
});
$(”#btnJump”).click(function(){
        page=$(”input”).val();
        $.post(”/CI/index.php/acemap/new_result_paper_page
            ”,{”title”:title,”page”:page},function(data,
            status){
                $(”table”).replaceWith(data);

        })
});
```

When we click the button, it call the *new_result_paper_page* function. This function retrieve ten data based on the page number sent.

```
public function new_result_paper_page()
{
    $title=$this->input->post(”title”);
    $page=$this->input->post(”page”);
    $data['items']=$this->ace_model->get_some_title($title,
        $page);
    $this->load->view(”acemap/new_result_paper_page”,$data);
}
```

And the function return a page, only containing the table part, so that the original page can replace only the table with the new one. As the following shows



Title	Citation	Year
nonlinear mappings with cellular neural networks	0	2005
online human action detection using joint classification regression recurrent neural networks	0	2016
a study of the application of the cmac artificial neural network to the problem of gas sensor array calibration	0	1995
an integrated architecture of adaptive neural network control for dynamic systems	0	1995
fuzzy control and neural networks applications for consumer products	0	1991
quadratic type lyapunov functions for competitive neural networks with different time scales	0	1996
analysis of deep neural networks with extended data jacobian matrix	0	2016
extracting propositional rules from feed forward neural networks a new decompositional approach	0	2007
a neural network that learns to interpret myocardial planar thallium scintigrams	0	1993
decoding 1 d barcode from degraded images using a neural network	0	2010

Figure 7: Page 29 of the result of ”neural network”.

2.4 UI design

This part we will introduce the process of beautifying the website.

2.4.1 Home Page

Here we first change the position of the search box to the center of the page.

```
#header {
    color: #003366;
    font-family: ”Arial”;
        font-size: 50px;
    font-weight: bold;
    font-style: italic;
    text-align:center;
    margin-top: 15px;
    padding:5px;
    padding-top: 0px;
}
```

Then we use this id:

```
<div id='header'>
                <h1 style=”font-size: 50px;”><?php echo $title; ?></h1>
                <h2><?php if(isset($subtitle)){echo $subtitle;} else{echo ””;} ?></h2>
        </div>
        <div id='section' class=”greenout”>
        <p id='info'>Enter the scholar's name</p>
```

Next we add border and color to the box:

```
.greenout
{
text-align: center;
background-color:#006666;
color:white;
font-weight: bold;
outline-style: outset;
outline-color:#006666;
outline-width:10px;
}
```

We also use bootstrap to decorate the button:

```
<linkrel=”stylesheet” href=”https://cdn.bootcss.com/bootstrap/3.3.7/css/bootstrap.min.css”>
        <script src=”https://cdn.bootcss.com/jquery/2.1.1/jquery.min.js”></script>
        <script src=”https://cdn.bootcss.com/bootstrap/3.3.7/js/bootstrap.min.js”></script>
<input type=”submit” name=”submit” value=”Search” id=”submit_btn” class=”btn btn-success”/>
```

Then we give the overall code (with the above codes combined). Note that with MVC pattern, the style is added to the *header.php* and don't need to repeat in every view page.

```
<style type=”text/css”>
.greenout
{text-align: center;
background-color:#006666;
color:white;
font-weight: bold;
outline-style: outset;
outline-color:#006666;
outline-width:10px;
}
h1
{
font-size: 50px;
}
div.homepage
{
text-align: center;
background-color:#CC9900;
color:#330099;
outline-style: outset;
outline-color:#CC9900;
outline-width:10px;
position:absolute;
left:30%;
right:30%;
top:200px;
font-style:italic;
font-size:30px;
font-weight:bold;
}
table,th,td
{
text-align: center;
}
td
{
border-top-style: solid;
border-top-color: #0099FF;
border-top-width: 1px;
}
</style>
<center>
<div id='all'>
        <div id='header'>
                <h1 style=”font-size: 50px;”><?php echo $title; ?></h1>
                <h2><?php if(isset($subtitle)){echo $subtitle;} else{echo ””;} ?></h2>
        </div>
        <div id='section' class=”greenout”>
        <p id='info'>Enter the scholar's name</p>
        <form action=”” method=”post” id=”myform”>

            <input type=”input” id=”tags” name=”authorname” style=”color: black;” />
            <input type=”submit” name=”submit” value=”Search” id=”submit_btn” class=”btn btn-success”/>

            <br><br>
            <input type=”radio” name=”check” id=”au” checked> Author
```

```
                    <input type=”radio” name=”check” id=”pa”>
                        Paper
                    <input type=”radio” name=”check” id=”co”>
                        Conference

            </form>
            </div>
</div></center>;
```

Then we get the changed page:

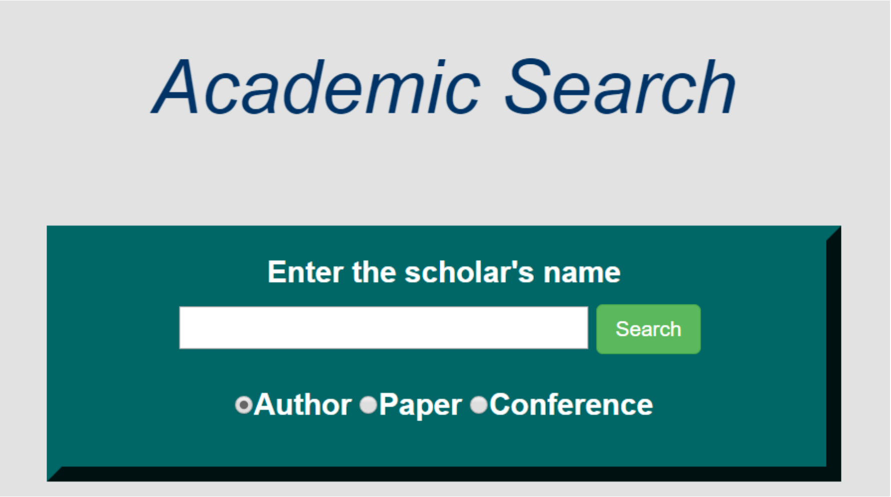


Figure 8: The refined homepage.

2.4.2 Result Page

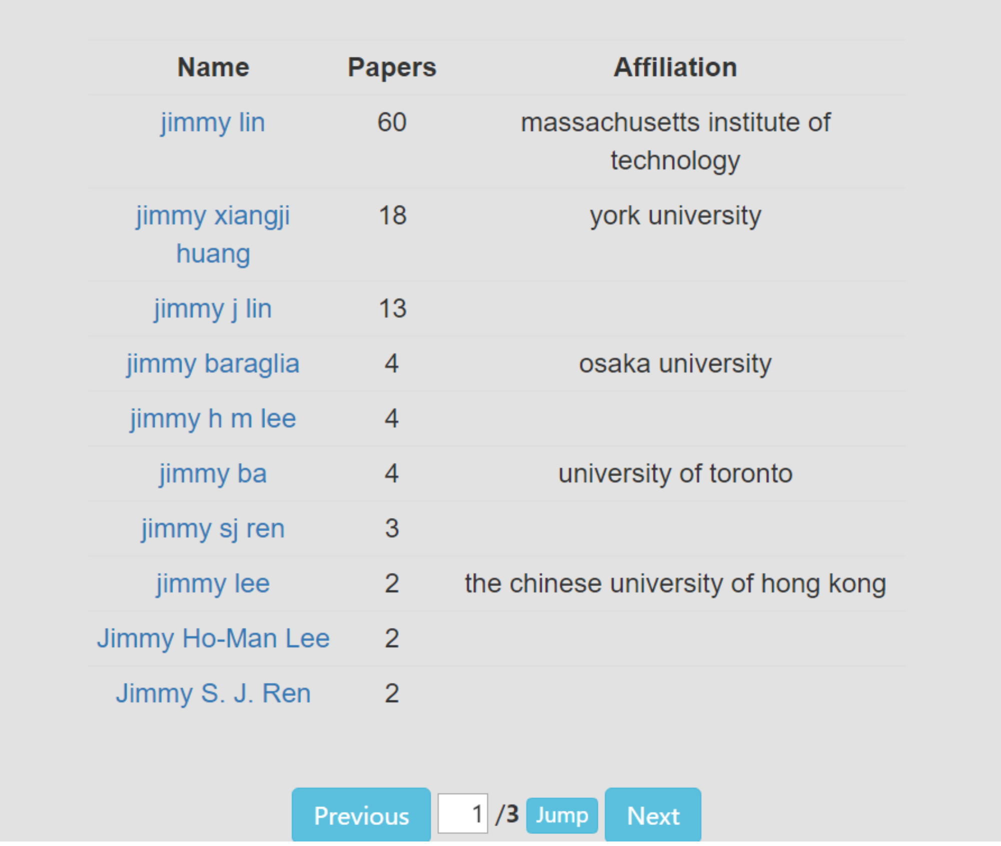
On this page we show the result of the searching. We put the table on the center:

```
#header {
    color: #003366;
    font-family: ”Arial”;
        font-size: 50px;
    font-weight: bold;
    font-style: italic;
    text-align:center;
    margin-top: 15px;
    padding:5px;
    padding-top: 0px;
}
#all {
        position: absolute;
        left:30%;
    right:30%;
    margin-top: 0px;
}
```

Here we also use bootstrap to modify the button “Previous” and “Next”.We also add “Jump” button.

```
<button id=’btnPre’ class=”btn btn−info”>Previous</button>
<input type=”text” style=”width:30px;text−align:right” name=”page” value=1>
    /<label><?=$total?></label>
<button id=’btnJump’ class=”btn btn−info btn−xs”>Jump</button>
<button id=’btnNext’ class=”btn btn−info”>Next</button>
```

Then we give get page like:

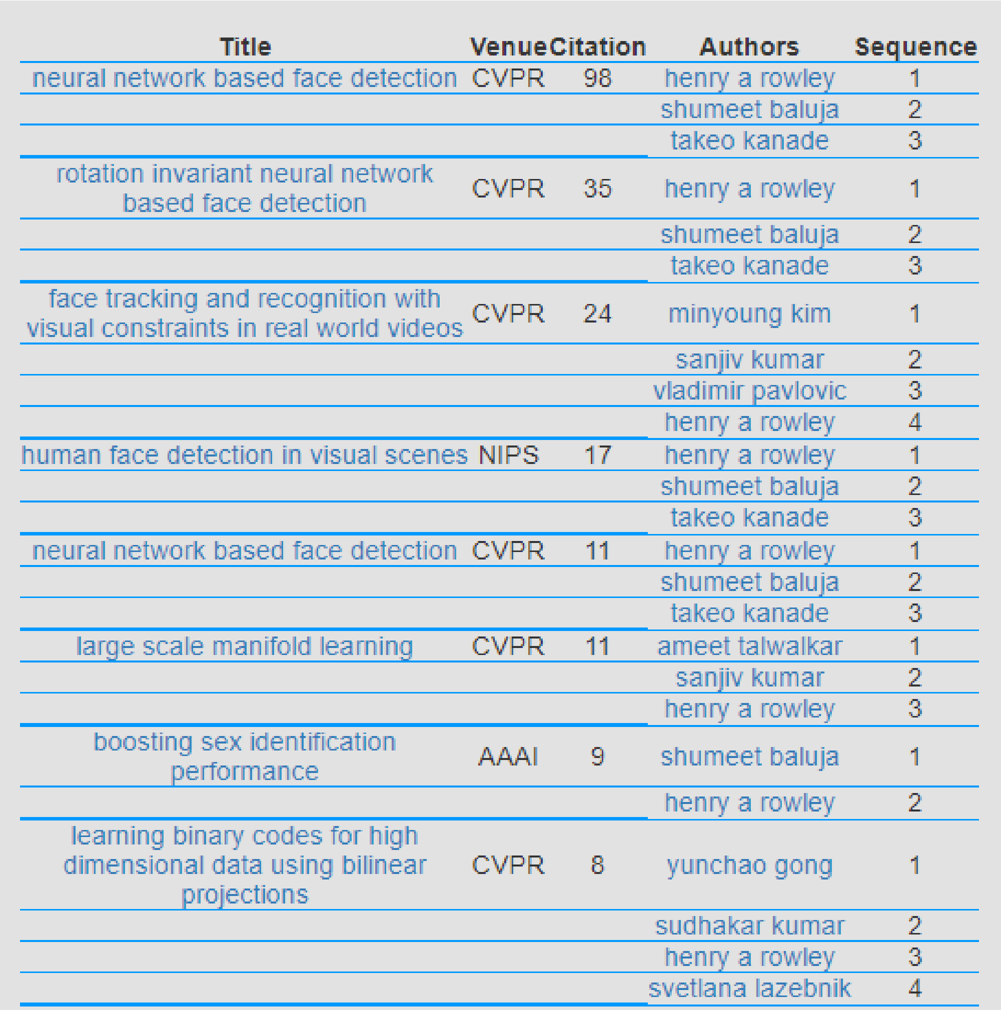


Name	Papers	Affiliation
jimmy lin	60	massachusetts institute of technology
jimmy xiangji huang	18	york university
jimmy j lin	13	
jimmy baraglia	4	osaka university
jimmy h m lee	4	
jimmy ba	4	university of toronto
jimmy sj ren	3	
jimmy lee	2	the chinese university of hong kong
Jimmy Ho-Man Lee	2	
Jimmy S. J. Ren	2	

Figure 9: The refined result page.

2.4.3 Author page

On this page we also use the table property and bootstrap to modify the button.We also add the visualizations to reflect the mentor-student relationship. For the table part, we use css directly:



Title	Venue	Citation	Authors	Sequence
neural network based face detection	CVPR	98	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
rotation invariant neural network based face detection	CVPR	35	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
face tracking and recognition with visual constraints in real world videos	CVPR	24	minyoung kim	1
			sanjiv kumar	2
			vladimir pavlovic	3
			henry a rowley	4
human face detection in visual scenes	NIPS	17	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
neural network based face detection	CVPR	11	henry a rowley	1
			shumeet baluja	2
			takeo kanade	3
large scale manifold learning	CVPR	11	ameet talwalkar	1
			sanjiv kumar	2
			henry a rowley	3
boosting sex identification performance	AAAI	9	shumeet baluja	1
			henry a rowley	2
learning binary codes for high dimensional data using bilinear projections	CVPR	8	yunchao gong	1
			sudhakar kumar	2
			henry a rowley	3
			svetlana lazebnik	4

Figure 10: The refined author page.

For the mentor-relation graph

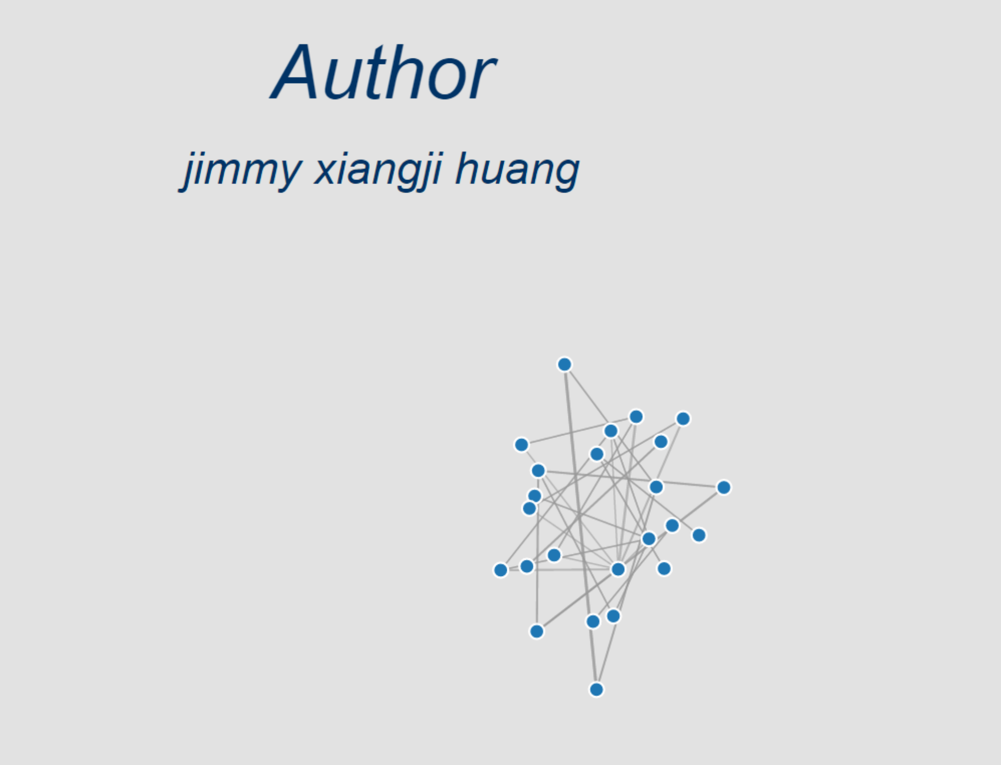


Figure 11: The refined relation-graph.

2.4.4 Menu

To further facilitate the use of the website, we add a toolbar. We change the background and some icons to make it look more nature and beautiful. We introduce an external css template. It can realize some animation and can hide the bar if not used. The following is the source code.

```
<link rel=”stylesheet” type=”text/css” href=”/css/nav.css”>
<link rel=”stylesheet” type=”text/css” href=”/font/iconfont.css”>

<script type=”text/javascript” src=”/js/jquery.min.js”></script>
<script type=”text/javascript” src=”/js/nav.js”></script>

<div id=”nav” class=”nav” style=”position: fixed;top: 0px;”>
    <div class=”nav–top”>
        <div id=”mini” style=”border–bottom:1px solid rgba(255,255,255,.1)”><img src=”/images/mini.png” ></div>
    </div>
    <ul>

        <li class=”nav–item”>
            <a href=”javascript:;”><i class=”my–
```

```
              icon nav−icon icon_2”></i><span>
              Navigation</span><i class=”my−
              icon nav−more”></i></a>
            <ul>
                <li><a href=”/CI/index.php/
                    acemap/home”><span>Main
                    Page</span></a></li>
            </ul>
        </li>
        <li class=”nav−item”>
            <a href=”javascript:;”><i class=”my−
              icon nav−icon icon_1”></i><span>
              Option</span><i class=”my−icon
              nav−more”></i></a>
            <ul>
                <li><a href=’’><span id=’del
                    ’>Delete History</span></
                    a></li>
            </ul>
        </li>
        <li class=”nav−item”>
            <a href=”javascript:;”><i class=”my−
              icon nav−icon icon_3”></i><span>
              History</span><i class=”my−icon
              nav−more”></i></a>
            <ul id=’history_content’>
                <li><a href=”javascript:;”><
                    span>No History</span></a
                    ></li>
            </ul>
        </li>

    </ul>
</div>
<script type=”text/javascript”>
    $.post(”/CI/index.php/acemap/get_history”,function(
      data,status){
        $(”#history_content”).replaceWith(data);
    });
    $(’#del’).click(function(){
        $.post(”/CI/index.php/acemap/del_history”,
          function(data,status){
        $(”#history_content”).replaceWith(data);
    });
    });
</script>
```

The effect is like this:

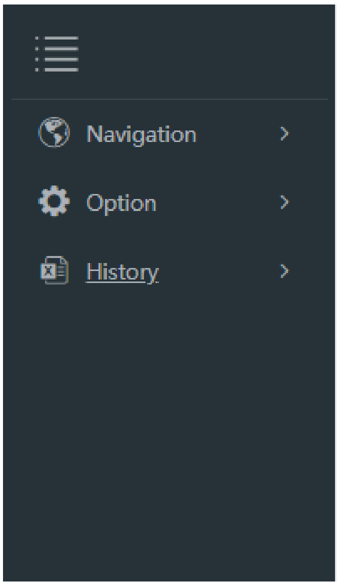


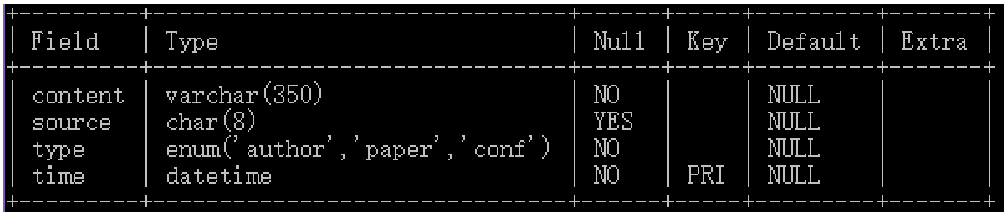
Figure 12: The sidebar.

3 New Features

3.1 History Record

Recording users’ browsing history has many benefits. Users may want to see what he just browsed, and turns to that page. The website can also use the history to analyse the perferance and makes good recommendations.

We used the mysql database to store the history. First, create a table called *userdata*. It stores the content of the history, the source of the history (to make hyperlink) and the exact time that the record comes into being.



```
+---------+-------------------------------+------+-----+---------+-------+
| Field   | Type                          | Null | Key | Default | Extra |
+---------+-------------------------------+------+-----+---------+-------+
| content | varchar(350)                  | NO   |     | NULL    |       |
| source  | char(8)                       | YES  |     | NULL    |       |
| type    | enum('author','paper','conf') | NO   |     | NULL    |       |
| time    | datetime                      | NO   | PRI | NULL    |       |
+---------+-------------------------------+------+-----+---------+-------+
```

Figure 13: The parameters of table userdata.

Then, for each function that load a specifi page, a SQL sentence is sent to add a new record. So we added a model function called *add_history*

```
public function add_history($content,$source,$type)
{
        $query=$this->db->query(”insert into userdata values
            (’$content’,’$source’,’$type’,now())”);
        return true;
}
```

Then, for the author page, paper page and conference page, we add records in the controller before it loads these pages. For example, in the controller function *authorPage*, we add:

```
$this->ace_model->add_history($data[’subtitle’],$authorid,’author’);
```

And for the paper page, we add:

```
$this->ace_model->add_history($data[’item’][’title’],$paperid,’paper’);
```

So, the history is recorded. Then, to load the existed record to the menu. we should search for the records and order them by time reversely.

In the model:

```
public function get_history()
{
        $query=$this->db->query(”select * from userdata order by time desc limit 20”);
        return $query->result_array();
}
```

In the menu page, for the history navigation bar:

```
<li class=”nav-item”>
        <a href=”javascript:;”><i class=”my-icon nav-icon icon_3”></i><span>History</span><i class=”my-icon nav-more”></i></a>
        <ul id=’history_content’>
                <li><a href=”javascript:;”><span>No History</span></a></li>
        </ul>
</li>
```

use javascript to call the get_history in the controller and retrieve data from menu_content page.

```
        $.post(”/CI/index.php/acemap/get_history”,function(data,status){
                $(”#history_content”).replaceWith(data);
        });
        $(’#del’).click(function(){
                $.post(”/CI/index.php/acemap/del_history”,function(data,status){
                $(”#history_content”).replaceWith(data);
        });
        });
```

In the controller:

```
public function get_history()
{
    $data[’items’]=$this->ace_model->get_history();
    $this->load->view(’acemap/menu_content’,$data);
}
```

And the menu_content.php

```
<ul id=’history_content’>
        <?php foreach ($items as $item): ?>
                <li><a href=
                        <?php
                        if($item[’type’]==’author’){echo ’/CI
                            /index.php/acemap/authorPage/’.
                            $item[’source’];}
                        else if($item[’type’]==’paper’){echo
                            ’/CI/index.php/acemap/paper_page
                            /’.$item[’source’];}
                        ?>
                        ><span><?=$item[’content’]?></span></
                            a></li>
        <?php endforeach; ?>
</ul>
```

To delete all the history, just a sql sentence:

```
$query=$this->db->query(”delete from userdata”);
```

The outcome is like this:

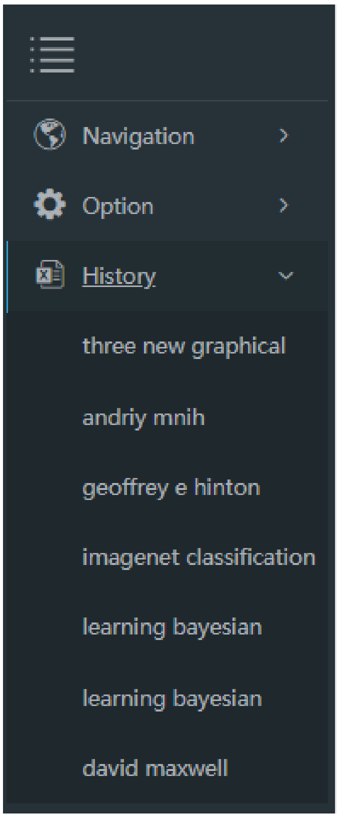


Figure 14: The history in the menu.

3.2 Paper Recommendation

Recommendation is a complicated procedure. Many factors need considering. For now, we only consider the relation. In detail, the website recommends papers that this paper cites from, or papers that cites this paper. And among are this candidates, the most poplular papers (more citation) is more likely to be favored by the user.

Add functions in the model to get the reference and citation of a particular paper.

```
public function get_ref($paperid)
{
        $query=$this->db->query(”select id,title from (select
             paper.id,paper.title from paper,reference where
            reference.paperid=’$paperid’ and paper.id=
            reference.refid)t1 left join reference on
            reference.refid=t1.id group by t1.id order by
            count(*) desc limit 4”);
        return $query->result_array();
}
public function get_cit($paperid)
{
        $query=$this->db->query(”select id,title from (select
             paper.id,paper.title from paper,reference where
            reference.refid=’$paperid’ and paper.id=reference
            .paperid)t1 left join reference on reference.
            refid=t1.id group by t1.id order by count(*) desc
             limit 2”);
        return $query->result_array();
}
```

In the controller, a function for the page to call.

```
public function recommendation()
{
    $paperid=$this->input->post(”paperid”);
    $data[’refs’]=$this->ace_model->get_ref($paperid);
    $data[’cits’]=$this->ace_model->get_cit($paperid);
    $this->load->view(’acemap/recommendation’,$data);
}
```

Then in the paper page, add a container.

```
<h3>You may be interested in:</h3>
<ul id=’recom’>
        <li></li>
</ul>
```

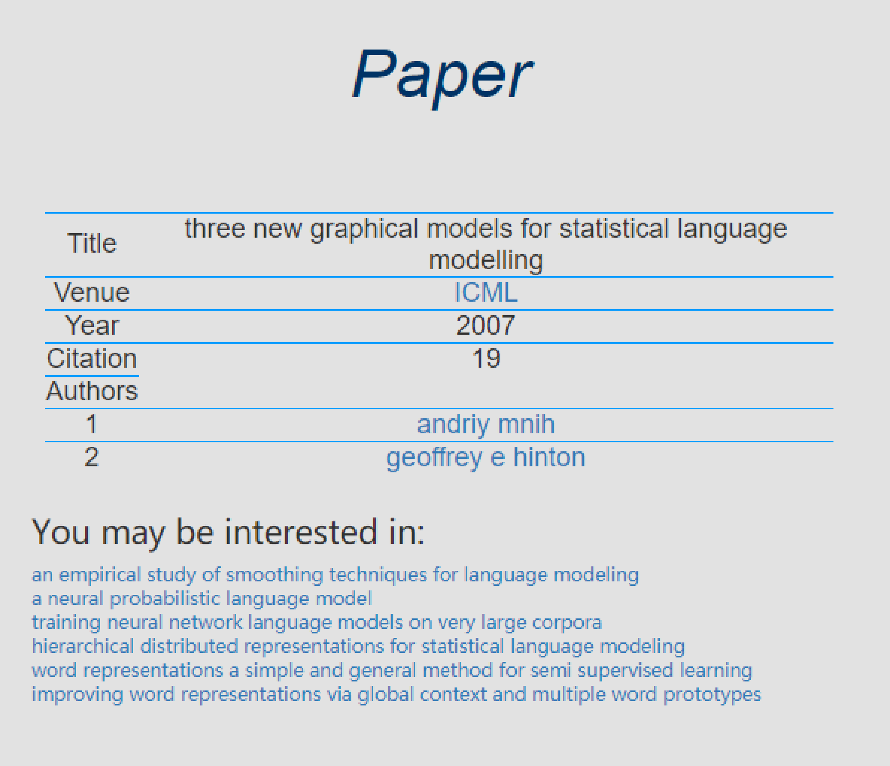
Then js scripts to fill it.

```
var paperid=$(”h2”).text();
$.post(”/CI/index.php/acemap/recommendation”,{”paperid”:
    paperid},function(data,status){
        $(”#recom”).replaceWith(data);
});
```

For the call-back page *recommendation.php*

```
<ul id=’recom’>
        <?php foreach ($refs as $item): ?>
        <li><a href=”/CI/index.php/acemap/paper_page/<?=$item
            [’id’]?>”><?=$item[’title’]?></a></li>
        <?php endforeach; ?>
        <?php foreach ($cits as $item): ?>
        <li><a href=”/CI/index.php/acemap/paper_page/<?=$item
            [’id’]?>”><?=$item[’title’]?></a></li>
        <?php endforeach; ?>
</ul>
```

The outcome is like this:



Paper

Title	three new graphical models for statistical language modelling
Venue	ICML
Year	2007
Citation	19
Authors	
1	andriy mnih
2	geoffrey e hinton

You may be interested in:

an empirical study of smoothing techniques for language modeling
a neural probabilistic language model
training neural network language models on very large corpora
hierarchical distributed representations for statistical language modeling
word representations a simple and general method for semi supervised learning
improving word representations via global context and multiple word prototypes

Figure 15: Paper recommendation.

4 Possible Improvements

4.1 Login-in Mechanism

For different users, they will look different pages and produce different data. Now, our website cannot separate it. So it is necessary to embed a login-in mechanism. Every user to brownse this website, must login in. If they don't have the account, they must sign up. To realize it, we can follow the procedure:

1. create a table for each user when they sign up
2. create a table to store all the users' accounts and passwords.
3. insert history record to different table of the current user.

4.2 Smarter Recommendation

A recommender system or a recommendation system (sometimes replacing ”system” with a synonym such as platform or engine) is a subclass of information filtering system that seeks to predict the ”rating” or ”preference” a user would give to an item. Our current recommendation system is not good enough. There are some other ways of recommendation algorithms.

Collaborative filtering

One approach to the design of recommender systems that has wide use is collaborative filtering.

Collaborative filtering methods are based on collecting and analyzing a large amount of information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an ”understanding” of the item itself.

When building a model from a user's behavior, a distinction is often made between explicit and implicit forms of data collection.

Examples of explicit data collection include the following:

- Asking a user to rate an item on a sliding scale.
- Asking a user to search.
- Asking a user to rank a collection of items from favorite to least favorite.
- Presenting two items to a user and asking him/her to choose the better one of them.
- Asking a user to create a list of items that he/she likes.

Examples of implicit data collection include the following:

- Observing the items that a user views in an online store.
- Analyzing item/user viewing times.[35]
- Keeping a record of the items that a user purchases online.
- Obtaining a list of items that a user has listened to or watched on his/her computer.
- Analyzing the user's social network and discovering similar likes and dislikes.

The recommender system compares the collected data to similar and dissimilar data collected from others and calculates a list of recommended items for the user. Several commercial and non-commercial examples are listed in the article on collaborative filtering systems.

One of the most famous examples of collaborative filtering is item-to-item collaborative filtering (people who buy x also buy y), an algorithm popularized by Amazon.com's recommender system.

Content-based filtering

Another common approach when designing recommender systems is content-based filtering. Content-based filtering methods are based on a description of the item and a profile of the user's preferences. In a content-based recommender system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. This approach has its roots in information retrieval and information filtering research.

To abstract the features of the items in the system, an item presentation algorithm is applied. A widely used algorithm is the tf–idf representation (also called vector space representation).

To create a user profile, the system mostly focuses on two types of information:

1. A model of the user's preference.
2. A history of the user's interaction with the recommender system.

Basically, these methods use an item profile (i.e., a set of discrete attributes and features) characterizing the item within the system. The system creates a content-based profile of users based on a weighted vector of item features. The weights denote the importance of each feature to the user and can be computed from individually rated content vectors using a variety of techniques. Simple approaches use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian Classifiers, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability that the user is going to like the item.

Direct feedback from a user, usually in the form of a like or dislike button, can be used to assign higher or lower weights on the importance of certain attributes (using Rocchio classification or other similar techniques).

Hybrid recommender systems

Recent research has demonstrated that a hybrid approach, combining collaborative filtering and content-based filtering could be more effective in some cases. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa); or by unifying the approaches into one model. Several studies empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrate that the hybrid methods can provide more accurate recommendations than pure approaches. These methods can also be used to overcome some of the common problems in recommender systems such as cold start and the sparsity problem.