GROUP 17

FINAL PROJECT

June 24, 2018

Chenyu Yang 517030910386
Yuting Lan 517030910376
Zihan Xu 517030910385
Dongwei Xie 517030910354

Contents

1 INTRODUCTION (DONGWEI XIE)

Project introduction: We have both compulsory tasks and optional tasks. For compulsory tasks, we ought to make it able to search with ability to do fuzzy search for authors, papers and conferences; show results and add turning-page buttons to display the result that is more than 10.For optional tasks, we need to optimize our website. We can design both layout and source codes to make it user-friendly and beautiful.

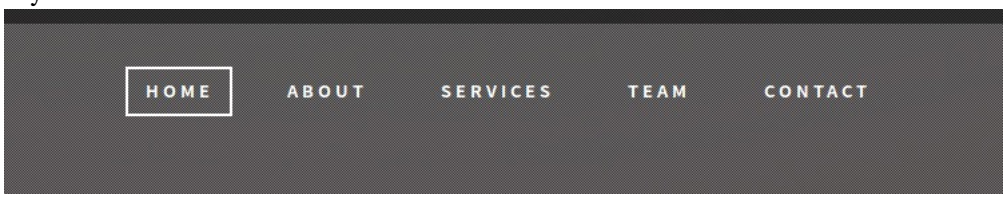
2 UI DESIGN (ZIHAN XU)

2.1 Page Design

The home page is designed for the presentation of all our functions. The origin page is based on a website template that is accessible from the Internet, so my job is to modify it and try to attach our desired function to the main page. Here are the core parts I want to implement in the main page:

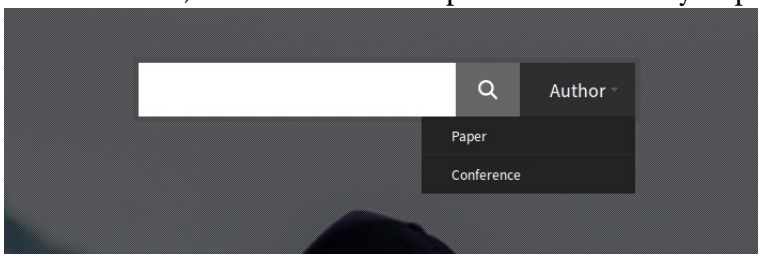
(1) Navigation Bar

The navigation bar consists of buttons that lead the user to the corresponding part of the page, which are: HOME: directed to the home page, which is actually itself; ABOUT: the introduction of the IEEE, SJTU, ACM and of course, IEEE pilot class; SERVICES: the introduction to all the services in our project; TEAM: the introduction to our team; CONTACT: the way to contact our team.



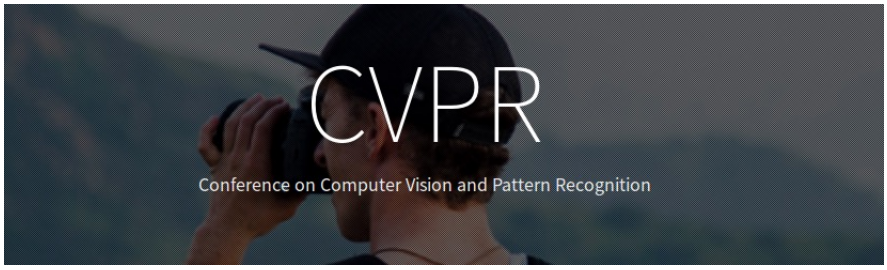
(2) The searching box

The searching box must be able to switch between author-search, paper-search and conference-search, so I wrote a css script and successfully implement the searching function.



(3) The scrolling bar

Under the searching bar is a scrolling bar, showing some main conferences in the academic field. You can click on the little pot in it and switch to different conferences easily.



2.2 Some problems

(1) different size:

When viewed from small screen (like screen of smart phone), the page became twisted, the position of the different box changed, and even worse, some sontent could not be shown properly. For instance, the searching bar could not work at all!

The source of the problem turned out to be the measuring way of the 'width' and 'height': when using units like 'px', 'em', the page performs differently in different divices. So my solution is simple but useful: transforming the data into % format:

```
<div style="background-position: 50% 50%;"></div>
```

As a result, the page shows the same in different devices.

(2) Slow Loading:

When testing the website, we found that the required loading time is much longer than we had expected. With the help of console, I noticed that JQuery took a long time to load. By analysis, the core problem bacame clear: we used JQuery from a foreign source, which is hard to have access to. Our soluiton is simple, we designed to use another source in China, and the problem just disappeared.

3 ELASTIC-SEARCH (ZIHAN XU)

3.1 what is elasticsearch

Elasticsearch is an open-source, broadly-distributable, readily-scalable, enterprise-grade search engine. Accessible through an extensive and elaborate API, Elasticsearch can power extremely fast searches that support your data discovery applications.

It is easy to get going with Elasticsearch. It ships with sensible defaults and hides complex search and distribution mechanics from beginners. It works quite well, right out of the box. With a short learning curve for grasping the basics, you can become productive very quickly.

3.2 why using elasticsearch

(1) Fast, Incisive Search against Large Volumes of Data

Conventional SQL database managements systems aren't really designed for full-text searches, and they certainly don't perform well against loosely structured raw data that resides outside the database. On the same hardware, queries that would take more than 10 seconds using SQL will return results in under 10 milliseconds in Elasticsearch.
(2) Indexing Documents to the Repository

During an indexing operation, Elasticsearch converts raw data such as log files or message files into internal documents and stores them in a basic data structure similar to a JSON object. Each document is a simple set of correlating keys and values: the keys are strings, and the values are one of numerous data types—strings, numbers, dates, or lists.
(3) Denormalized Document Storage: Fast, Direct access to your Data

It's important to remember that Elasticsearch isn't a relational database, so DBMS concepts usually won't apply. The most important concept that you must set aside when coming over from conventional databases is normalization. Native Elasticsearch doesn't permit joins or subqueries, so denormalizing your data is a essential.
(4) Broadly Distributable and Highly Scalable

Elasticsearch can scale up to thousands of servers and accommodate petabytes of data. Its enormous capacity results directly from its elaborate, distributed architecture. And yet the ES user can be thankfully unaware of nearly all of the automation and complexity that supports this distributed design.

3.3 the usage of elasticsearch

(1) installing the Elasticsearch

Since I am using elasticsearch in Ubuntu, the installation becomes quite easy.

As we know, Elasticsearch is based on Java, we have to update Java first.

```
$ sudo apt-get update
$ sudo apt-get upgrade

$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update

$ sudo apt-get install oracle-java7-installer
```

Then get access to the official source:

```
$ wget -O - http://packages.elasticsearch.org/GPG-KEY-elasticsearch | apt-key add
    -
$ sudo echo "deb http://packages.elasticsearch.org/elasticsearch/2.3.3.0/debian
    stable main" >> /etc/apt/sources.list

$ sudo apt-get update
$ sudo apt-get install elasticsearch
```

Now using the following command to start Elasticsearch

```
$ sudo /etc/init.d/elasticsearch start
```

(2) importing and indexing the data

There are two sources of data: the original data stored in the .txt file, and the data stored in the mysql database. Considering that we had made some adjustion to the original data, I chose to import the data from the second source.

In order to build connection with mysql database, we need to use a special tool: elasticsearch-jdbc.

First we need to download it:

```
$ wget http://xbib.org/repository/org/xbib/elasticsearch/importer/
    elasticsearch-jdbc/2.3.1.0/elasticsearch-jdbc-2.3.1.0-dist.zip
$ unzip elasticsearch-jdbc-2.3.3.0-dist.zip
```

The core part is to give commmand to elasticsearch-jdbc, introducing it to get the data from mysql.

So I have to write a script for elasticsearch-jdbc.

I will take author_db as an example.

```
import_paper.sh:

bin=/home/es/cluster/elasticsearch-2.3.1/elasticsearch-jdbc-2.3.3.0/bin
lib=/home/es/cluster/elasticsearch-2.3.1/elasticsearch-jdbc-2.3.3.0/lib
echo '{
"type" : "jdbc",
"jdbc": {
"url":"jdbc:mysql://localhost:3306/main_db",
"user":"root",
"password":"xzh981001",
"sql":"select * from Authors",
"index" : "Authors",
"type" : "info"
}}' | java \
    -cp "${lib}/*" \
    -Dlog4j.configurationFile=${bin}/log4j2.xml \
    org.xbib.tools.Runner \
    org.xbib.tools.JDBCImporter
```

Before executing the script, permission needs to be given to it:

```
chmod a+x improt_author.sh
```

Finally we can apply this script:

```
./import_author.sh
```

And all the data stored in the main_db.authors will be conveyed to the Elasticsearch server.

3.4 Replacing mysql with Elasticsearch

As has been shown, Elasticsearch is highly dependent on data with json format, the searching job is simply the same. I will use a simple example to explain how to replace mysql phrase with Elasticsearch.

In sql, the command to search an author whose id is '00153BD3', we can use:

```
mysql> select * from authors where authorid = 00153BD3;
```

The result is:

```
+----+-------------------------+
| authorid | authorname |
+----+-------------------------+
| 00153BD3 | matteo vasirani |
+----+-------------------------+
```

In Elasticsearch, the similar phrase is:

```
# curl http://localhost:9200/authors/info/_search?pretty -d '
{
"filter" : { "term" : { "authorid" : "00153BD3" } }
}'

And the result would be:

{
  "took" : 3,
  "timed_out" : false,
  "_shards" : {
  "total" : 8,
  "successful" : 8,
  "failed" : 0
  },
  "hits" : {
  "total" : 1,
  "max_score" : 1.0,
  "hits" : [ {
  "_index" : "authors",
  "_type" : "info",
  "_id" : "AVVXKgeEun6ksbtikOWJ",
  "_score" : 1.0,
```

```
        "_source" : {
25      "authorid" : 00153BD3,
        "authorname" : "matteo vasirani"
27      }
        } ]
29      }
      }
```

Then with php we can get the data I want conveniently.

Besides, in the searching part we would have to use fuzzy search to find the matches of names. In the mysql we could use % to use fuzzy search, and with it we will be able to search the authors' name — and also the full text search.

Here is the replacement of sql phrase:

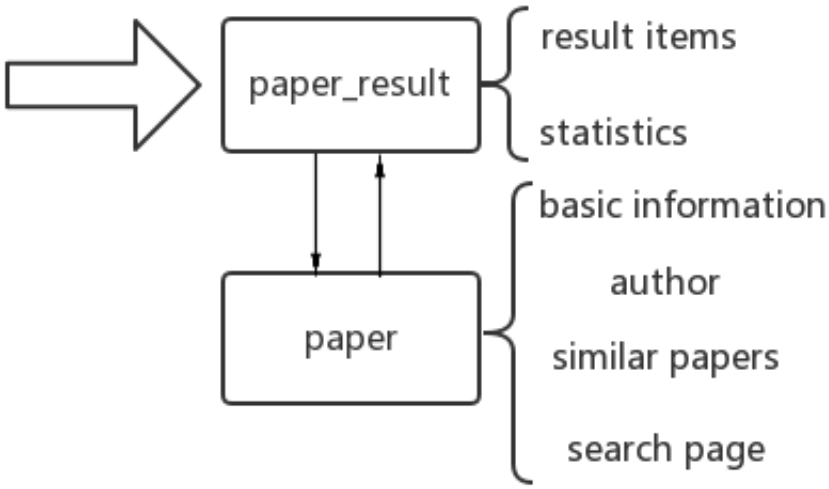
```
      GET /authors/info/_search
2     {
        "query": {
4         "match": {
            "text": {
6             "query":     "name",
              "fuzziness": "AUTO",
8             "operator":  "and"
            }
10        }
        }
12    }
```

And we can get the fuzzy result.

4 Paper and Recommendation (Chengyu Yang)

In this part,I have designed 2 php files: **p_result.php** shows the search result; **paper.php** displays the details of each paper.And I used web template to beautify the pages.

Here is the flow diagram of this part:



Firstly, we need to import related packages to ensure the page works.

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

<head>
    <meta charset="utf-8">

    <script type="text/javascript" src="jquery-3.3.1.min.js"></script>
    <script type="text/javascript" src="jquery.js"></script>

    <!-- Bootstrap Core CSS -->
    <link rel="stylesheet" type="text/css" href="assets/bootstrap/css/bootstrap.min.
        css">

    <!-- Font Awesome CSS -->
    <link rel="stylesheet" type="text/css" href="assets/css/font-awesome.min.css">


    <!-- Animate CSS -->
    <link rel="stylesheet" type="text/css" href="assets/css/animate.css">

    <!-- Owl-Carousel -->
    <link rel="stylesheet" type="text/css" href="assets/css/owl.carousel.css" >
    <link rel="stylesheet" type="text/css" href="assets/css/owl.theme.css" >
    <link rel="stylesheet" type="text/css" href="assets/css/owl.transitions.css" >

    <!-- Materialize CSS -->
    <link rel="stylesheet" type="text/css" href="assets/css/material.css">


    <!-- Custom CSS -->
    <link rel="stylesheet" type="text/css" href="assets/css/style.css">
    <link rel="stylesheet" type="text/css" href="assets/css/responsive.css">

    <!-- Colors CSS -->
```

```
    <link rel="stylesheet" type="text/css" href="assets/css/color/blue.css" title="
        blue">

    <!-- Modernizer js -->
    <script src="assets/js/modernizr.custom.js"></script>

    <?php
    if ($_GET["name"]=="")
        echo "<script>history.back(-1);</script>";
    $name=$_GET["name"];
    echo"<script>var jsname="."'$name'</script>";
    ?>

</head>
```

4.1 p_result.php

In this part, I put the selecing process in a backend named **p_result_back.php**.Only when the frontend gets the keyword,it will pass the word to the backend. The backend selects the title in the SQL ,then change the result into JSON data and return it to the frontend.The frontend will complete other functions (like sorting).

Here is the code of the backend:

```
<?php
include_once ("connect.php");


$inf=$_GET["name"];
$final_res=array();
$IDs=mysqli_query($conn,"SELECT a.PaperID, a.Title, a.PaperPublishYear, c.
    ConferenceName,a.ConferenceID,COUNT(f.PaperID)
                                    FROM papers a
                                    LEFT JOIN paper_reference f ON a.PaperID=f.
                                        ReferenceID
                                    LEFT JOIN conferences c ON a.ConferenceID=c.
                                        ConferenceID
                                    WHERE a.Title LIKE '%$inf%'
                                    GROUP BY a.PaperID
                                    ORDER BY COUNT(f.PaperID) DESC;");
while ($row=mysqli_fetch_assoc($IDs))
{
    $final_res[]=json_encode($row); // change the multiple array into JSON data
}
echo json_encode($final_res);
?>
```

The returned JSON data includes each items' Title, PaperID, PublishYear, Conference and cited times.And these are all the frontend needs.

Now we should consider how to display the data in a nice-looking way.The followings are one section of the front end which just shows the result:

```
<section id="result" class="services-section">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="section-title text-center">
                    <h2>Result</h2>
                    <p>We've found <span id="num"></span> papers!</p>
                </div>
            </div>
        </div>
        <div style="float: right;">
            <form action="">
                <select id="rank">
                    <option id="2" selected>Citations</option>
                    <option id="3">Year</option>
                </select>
            </form>
        </div>
        <div id="tab">
  <!--The result table  -->
        </div>
    </div>
    <center>

        <div><button class="btn btn-info" id="0">Previous</button>
            <span id="page"></span>
            <button class="btn btn-info" id="1">Next</button></div>


    </center>
</section>
```

In the span(id=**'num'**) ,I put the amount number of selecting results. Then in the div(id=**'tab'**),is a table of all the items. Users can turn the page just by click the two buttons(id=**'0'** and **'1'**).The span(id=**'page'**)shows the current page. Apart from this,users can change different orders(year and cited times) just by choosing the form(id=**'rank'**,2 options id=**'2'** and **'3'**).

This is only a basic frame.Then I use the JS to complete the functions and import the JSON data from the backend.

```
<script>
```

```
function write_table(i,result) {
    var content="<table border=\"1\" align=\"center\" style='width:100%;'> <
        th>Title</th><th>PublishYear</th><th>Citations</th><th>Conference</th>
        ";
    for (var j=0;j<10 && (i-1)*10+j<result.length;j++)
    {
        var item=eval("("+result[(i-1)*10+j]+")");
        content=content+"<tr><td><a href=\"http://localhost/paper.php?&id="+
            item["PaperID"]+"&submit=submit\">"+item["Title"]+"</a></td><td>"+
            item["PaperPublishYear"]+"</td><td>"+item["COUNT(f.PaperID)"]+"</
            td><td><a href=\"http://localhost/conference.php?id="+item["
            ConferenceID"]+"&submit=submit\">"+item["ConferenceName"]+"</a></
            td></tr>;";

    };
    content=content+"</table>";
    document.getElementById("tab").innerHTML=content;
    document.getElementById("page").innerHTML=i;
};
```

eval("("+JSON+")") is used to read the JSON data.

This function ***write_table(i,result)*** is used to write the data in the table(id=**'tab'**) .Two defaults: *i* is the current page nunmber and *result* is the JSON got from the backend including all the information.

When the table is changed-the page finished loading or that users turned another page-the ***write_table(i,result)*** function will be invoked.

```
$(document).ready(function(){
    var i=1;

    $.getJSON("p_result_back.php",{name:jsname},function(result){
        write_table(i,result);
        document.getElementById("num").innerHTML=result.length;

        $("#0").hide();
        if (result.length<=10)
            $("#1").hide();


        $("#0").click(function(){  \\click the PREVIOUS button
            $("#1").show();
            i--;
            if (i==1) \\adjust whether it's the first page
                $("#0").hide();
            write_table(i,result);
        });

        $("#1").click(function(){  \\click the NEXT button
            $("#0").show();
```

```
                    i++;

                    if (i*10>=result.length) \\adjust whether it's the last page
                        $("#1").hide();
                    write_table(i,result);
                });
```

In this part it get the data from the backend,and count the amount of records put the number in the span(**'num'**) and write the first 10 records.For this is the first page,the PREVIOUS button is hidden.And so is the NEXT button when it turns to the last page.

```
        function compare(property) {
            return function(a, b) {
                var value1 = eval("("+a+")")[property];
                var value2 = eval("("+b+")")[property];
                return  value2-value1;
            };
        };

                $("#2").click(function(){
                    i=1;
                    $("#0").hide();
                    $("#1").show();
                    result= result.sort(compare("COUNT(f.PaperID)"));
                    write_table(i,result);
                });

                $("#3").click(function(){
                    i=1;
                    $("#0").hide();
                    $("#1").show();
                    result= result.sort(compare("PaperPublishYear"));
                    write_table(i,result);
                });
            });
  });
    </script>
```

This part is in charge of sorting the records by the way of the ***compare(property)***.

When the option **'citations'**(id=**'2'**) is clicked,the records will be sorted in the order of cited times.And when the option **'years'**(id=**'3'**) is clicked,the records will be sorted in the order of their publish years.

Apart from this, I also designed a dynamic part to show all the statistical data.

```
    <section class="fun-facts">
        <div class="container">
            <div class="row">
```

```
                <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="counter-item waves-effect">
                      <i class="fa fa-male"></i>
                      <div class="timer" id="item1" data-to="113398" data-speed="
                          3000"></div>
                      <h5>authors</h5>
                    </div>
                  </div>
                  <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="counter-item waves-effect">
                      <i class="fa fa-paper-plane"></i>
                      <div class="timer" id="item2" data-to="90403" data-speed="
                          3000"></div>
                      <h5>papers</h5>
                    </div>
                  </div>
                  <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="counter-item waves-effect">
                      <i class="fa fa-book"></i>
                      <div class="timer" id="item3" data-to="3159" data-speed="3000
                          "></div>
                      <h5>affiliations</h5>
                    </div>
                  </div>
                  <div class="col-xs-12 col-sm-6 col-md-3">
                    <div class="counter-item waves-effect">
                      <i class="fa fa-home"></i>
                      <div class="timer" id="item4" data-to="13" data-speed="3000"
                          ></div>
                      <h5>conferences</h5>
                    </div>
                  </div>
            </div>
        </div>
    </section>
```

4.2 paper.php

This page is used to show the details of each paper.It includes four parts,so firstly I designed an index which can jump to a certain part.

```
<div class="menu-wrap">
    <nav class="menu">
        <div class="icon-list">
            <a href="#home" class="logo page-scroll waves-effect">Navigator</a>
            <a href="#home" class="page-scroll waves-effect"><i class="fa fa-fw fa-
                comment-o"></i><span>Information</span></a>
            <a href="#authors" class="page-scroll waves-effect"><i class="fa fa-fw fa
                -user"></i><span>Authors</span></a>
```

```
            <a href="#similar" class="page-scroll waves-effect"><i class="fa fa-fw fa
                -bar-chart-o"></i><span>Similar</span></a>
            <a href="#search" class="page-scroll waves-effect"><i class="fa fa-fw fa-
                envelope-o"></i><span>Search</span></a>
        </div>
    </nav>
    <button class="close-button" id="close-button">Close Menu</button>
</div>
<button class="menu-button waves-effect" id="open-button">Open Menu</button>
```

Four different parts are :

- Title,PublishYear and Conference
- Authors and recommendation based on the same author
- recommendation based on the same citations
- search page

Title,PublishYear and Conference

frontend

```
<section class="header" id="home">
    <div class="container">
        <div class="intro-text" id="back_0">
        </div>
    </div>
</section>

<script>
    $.get("paper_back_0.php",{id:jsid},function (result)
    {
        document.getElementById("back_0").innerHTML=(result);
    });
</script>
```

This will get the title publish year and conference of the paper and present them in a veaeutiful design.

backend

```
<?php
$ID=$_GET["id"];

include_once ("connect.php");
if ($conn->connect_error)
{
    die("Failed to connect! " . $conn->connect_error);//check the connection.
```

```
}
else {
    $basicinf = mysqli_fetch_assoc(mysqli_query($conn, "SELECT a.Title,a.
        PaperPublishYear,b.ConferenceName,b.ConferenceID
                                    FROM papers a LEFT JOIN conferences b ON a.
                                        ConferenceID=b.ConferenceID
                                    WHERE a.PaperID='$ID'LIMIT 1;"));
    echo "<h1>" . $basicinf["Title"] . "    <br><span>" . $basicinf["PaperPublishYear
        "] . "</span></h1>";
    echo "<p>----Published in <a href='http://localhost/conference.php?id=" .
        $basicinf["ConferenceID"]."&submit=submit'>". $basicinf["ConferenceName"] . "
        </a></p>";
}
?>
```

Authors and recommendation based on the same author

frontend

```
    <div id="authors" class="about-us-section-1">
        <div class="container">
            <div class="row" id="back_1">
            </div>
        </div>
    </div>

<script>
    $.get("paper_back.php",{id:jsid},function (res) {
        document.getElementById("back_1").innerHTML=(res);

    });
</script>
```

backend

This part shows the author information.It's divided into two divisions.The left one is a table,which contains all the authors and is listed in the author sequence.And it also shows the affiliation where the paper is published.

```
<?php
$ID=$_GET["id"];

include_once ("connect.php");
if ($conn->connect_error)
{
    die("Failed to connect! " . $conn->connect_error);//check the connection.
}
else
{
    echo "<div class=\"col-md-8\"><div style='background: silver'>";
    $query = mysqli_query($conn, "SELECT a.AuthorSequence,b.AuthorName,a.AuthorID,c.
        AffiliationName
```

```
                                    FROM paper_author_affiliation a
                    LEFT JOIN affiliations c ON a.AffiliationID=c.AffiliationID
                      LEFT JOIN authors b ON a.AuthorID=b.AuthorID
                                    WHERE a.PaperID='$ID'
                                    ORDER BY a.AuthorSequence;");
    $row = mysqli_fetch_assoc($query);
    if (empty($row["AffiliationName"]))
        echo "<h1>No publishment</h1><h4> in any affiliation.</h4></div>";
    else
        echo "<h4>Published in </h4><h1>".$row["AffiliationName"]."</h1></div>";
  \\search the affiliation

    $author_paper=array(); \\Prepare for the recommendation based on authors

    echo "<div><table frame=\"above\" width=\"100%\" align='top'>";
    echo "<th>Author Sequence</th>","<th>Author Name</th><th>Main Affiliation</th>";
    echo "<tr><td>".$row["AuthorSequence"] ."</td><td><a href='http://localhost/
        author.php?id=".$row["AuthorID"]."&submit=submit'>". $row["AuthorName"] . "</a
        ></td>";
    $authorid=$row["AuthorID"];
    $Authordata=mysqli_query($conn,"SELECT b.AffiliationName,count(DISTINCT a.PaperID
        ) FROM paper_author_affiliation a
                                        LEFT JOIN  affiliations b ON a.
                                            AffiliationID=b.AffiliationID
                                        WHERE a.AuthorID='$authorid'
                                        GROUP BY a.AffiliationID ORDER BY count
                                            (DISTINCT a.PaperID) DESC limit 1;")
                                            ;
    $main_Affi=mysqli_fetch_assoc($Authordata)["AffiliationName"];
    if (empty($main_Affi))
        $main_Affi="None";
    echo "<td>".$main_Affi."</td></tr>";

    $papers=mysqli_query($conn,"SELECT b.Title ,b.PaperID,1/a.AuthorSequence*(1+count
        (c.PaperID)) as weight FROM paper_author_affiliation a
                                    LEFT JOIN papers b ON a.PaperID=b.PaperID
                                    LEFT JOIN paper_reference c ON a.PaperID=c.
                                        ReferenceID
                                    WHERE a.AuthorID='$authorid'
                                    GROUP BY a.PaperID ORDER BY weight DESC limit
                                        5;");
    $tmp=array();
    while($paper=mysqli_fetch_assoc($papers))
        $tmp[$paper["PaperID"]]=$paper["Title"];
    $author_paper[$row["AuthorName"]]=$tmp;


    while ($row = mysqli_fetch_assoc($query))
    {
        echo "<tr><td>".$row["AuthorSequence"] ."</td><td><a href='http://localhost/
            author.php?id=".$row["AuthorID"]."&submit=submit'>". $row["AuthorName"] .
            "</a></td>";
```

```
        $authorid=$row["AuthorID"];
        $Authordata=mysqli_query($conn,"SELECT b.AffiliationName,count(DISTINCT a.
            PaperID) FROM paper_author_affiliation a
                                            LEFT JOIN  affiliations b ON a.
                                                AffiliationID=b.AffiliationID
                                            WHERE a.AuthorID='$authorid'
                                            GROUP BY a.AffiliationID ORDER BY count
                                                (DISTINCT a.PaperID) DESC limit 1;")
                                                ;
        $main_Affi=mysqli_fetch_assoc($Authordata)["AffiliationName"];
        if (empty($main_Affi))
            $main_Affi="None";
        echo "<td>".$main_Affi."</td></tr>";

        $papers=mysqli_query($conn,"SELECT b.Title ,b.PaperID,1/a.AuthorSequence*(1+
            count(c.PaperID)) as weight FROM paper_author_affiliation a
                                        LEFT JOIN papers b ON a.PaperID=b.PaperID
                                        LEFT JOIN paper_reference c ON a.PaperID=c.
                                            ReferenceID
                                        WHERE a.AuthorID='$authorid'
                                        GROUP BY a.PaperID ORDER BY weight DESC limit
                                            4;");
        $tmp=array();
        while($paper=mysqli_fetch_assoc($papers))
            $tmp[$paper["PaperID"]]=$paper["Title"];
        $author_paper[$row["AuthorName"]]=$tmp;
    }
    echo "</table></div></div>  ";
```

The right part is recommendations with the same authors.If another paper of the author is widely refered,it is more likely to be important.And the author sequence means the relationship between the paper and the author.We want to know those both related and well-known paper,so these two factors are taken into consideration:

$$weight = (1 + cited - times)/authorsequence$$

```
    echo     " <div class=\"col-md-4\" style='height:500px; overflow: auto; float:
        right'>
                <div class=\"custom-accordion waves-effect\">
                    <!-- Start Accordion Section -->
                    <div class=\"panel-group\" id=\"accordion\">";
//  print_r($author_paper);
 $mark=1;
 foreach($author_paper as $name => $popular_papers) {
    echo "                          <!-- Start Accordion 1 -->
                        <div class=\"panel panel-default\">
                            <div class=\"panel-heading waves-effect\">
                                <h4 class=\"panel-title\">
```

```
                                        <a data-toggle=\"collapse\" data-parent=\"#
                                            accordion\" href=\"#collapse-$mark\">
                                            <i class=\"fa fa-angle-left control-icon\
                                                \"></i> $name
                                        </a>
                                    </h4>
                                </div>
                                <div id=\"collapse-$mark\" class=\"panel-collapse
                                    collapse in\">
                                    <div class=\"panel-body\">";
        $mark++;
        foreach ($popular_papers as $paper_id => $paper_title)
        {
            if ($paper_id == $ID)
                continue;
            echo " <a href='http://localhost/paper.php?id=$paper_id&submit=submit'>
                $paper_title</a><br><br>";
        }
        echo "</div></div>";
    }
    echo "</div></div></div>";
}
?>
```

recommendation based on the same citations

This part will recommend papers based on the same citation.I designed to show these recommendations in a way of sticky notes.And it can roll from left to right.

frontend

```
<section id="similar" class="client-section">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="section-title text-center wow fadeInDown" data-wow-
                    duration="2s" data-wow-delay="50ms">
                    <h2>Similar Papers</h2>
                    <p>Other papers with the same references:</p>
                </div>
                <div class="row">

                    <div class="col-md-12">
                        <div class="testimonial-section">
                            <div class="testimonial" id="q0">
                                <p id="t0" style="height: 230px;"> </p>
                                <div class="testimonial-people pull-right">
                                    <img src="assets/images/clients/client_4.png"
                                        class="img-responsive" alt="Testimonial People
                                        ">
                                </div>
                            </div>
                            <div class="testimonial" id="q1">
```

```
                    <p id="t1" style="height: 230px;"> </p>
                    <div class="testimonial-people pull-right">
                        <img src="assets/images/clients/client_4.png"
                            class="img-responsive" alt="Testimonial People
                            ">
                    </div>
                </div>
                <div class="testimonial" id="q2">
                    <p id="t2" style="height: 230px;"> </p>
                    <div class="testimonial-people pull-right">
                        <img src="assets/images/clients/client_4.png"
                            class="img-responsive" alt="Testimonial People
                            ">
                    </div>
                </div>
                <div class="testimonial" id="q3">
                    <p id="t3" style="height: 230px;"> </p>
                    <div class="testimonial-people pull-right">
                        <img src="assets/images/clients/client_4.png"
                            class="img-responsive" alt="Testimonial People
                            ">
                    </div>
                </div>
                <div class="testimonial" id="q4">
                    <p id="t4" style="height: 230px;"> </p>
                    <div class="testimonial-people pull-right">
                        <img src="assets/images/clients/client_4.png"
                            class="img-responsive" alt="Testimonial People
                            ">
                    </div>
                </div>
                <div class="testimonial" id="q5">
                    <p id="t5" style="height: 230px;"> </p>
                    <div class="testimonial-people pull-right">
                        <img src="assets/images/clients/client_4.png"
                            class="img-responsive" alt="Testimonial People
                            ">
                    </div>
                </div>
                <div class="testimonial" id="q6">
                    <p id="t6" style="height: 230px;"> </p>
                    <div class="testimonial-people pull-right">
                        <img src="assets/images/clients/client_4.png"
                            class="img-responsive" alt="Testimonial People
                            ">
                    </div>
                </div>
                <div class="testimonial" id="q2">
                    <p id="t2" style="height: 230px;"> </p>
                    <div class="testimonial-people pull-right">
                        <img src="assets/images/clients/client_4.png"
                            class="img-responsive" alt="Testimonial People
```

```
                                        ">
                                    </div>
                                </div>

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


            </div><!-- /.row -->
        </div><!-- /.container -->
</section>

<script>
    $(".testimonial").hide();
    $.getJSON("paper_back_2.php",{id:jsid},function(res){
        $.each(res,function(n,value) {
            var item=eval('('+value+')');
            $("#q" + n).show();
            document.getElementById("t"+n).innerHTML=("<div><span style='float: left
                '>Year:"+item["PaperPublishYear"]+"</span><span style='float: right'>
                Cited:"+item["count(a.ReferenceID)"]+"</span></div><br><br><center><
                div><font size='5'> <a href='http://localhost/paper.php?id="+item["
                PaperID"]+"&submit=submit'>"+item["Title"]+"</a></font></div></center>
                ")
        });
    });
</script>
```

The data is got in the way of JSON,which can be more convenient and flexible.It will write the title publishyear and cited times into each 'sticky note'.The user can draw them from left to right.If the number of items is less than 7.The rest blank notes will be hidden automatically.

The JSON data is like this:


```
{
    {"PaperID":"74C3FB40","Title":"what does classifying more than 10 000 image categories tell us","PaperPublishYear":"2010","count(a.ReferenceID)":"5"},
    {"PaperID":"80E668EA","Title":"multi level discriminative dictionary learning towards hierarchical visual categorization","PaperPublishYear":"2013","count(a.ReferenceID)":"4"},
    {"PaperID":"7D474E3D","Title":"what s it going to cost you predicting effort vs informativeness for multi label image annotations","PaperPublishYear":"2009","count(a.ReferenceID)":"4"},
    {"PaperID":"7AE55C40","Title":"learning hierarchical similarity metrics","PaperPublishYear":"2012","count(a.ReferenceID)":"4"},
    {"PaperID":"7D74B28A","Title":"max margin additive classifiers for detection","PaperPublishYear":"2009","count(a.ReferenceID)":"4"},
    {"PaperID":"7D180DF4","Title":"semantic kernel forests from multiple taxonomies","PaperPublishYear":"2012","count(a.ReferenceID)":"4"},
    {"PaperID":"794F4A06","Title":"photo recall using the internet to label your photos","PaperPublishYear":"2014","count(a.ReferenceID)":"3"},
    {"PaperID":"80122DFC","Title":"finding iconic images","PaperPublishYear":"2009","count(a.ReferenceID)":"3"},
    {"PaperID":"73E15989","Title":"scenenet a perceptual ontology for scene understanding","PaperPublishYear":"2014","count(a.ReferenceID)":"3"},
    {"PaperID":"7BCD9A60","Title":"evaluating knowledge transfer and zero shot learning in a large scale setting","PaperPublishYear":"2011","count(a.ReferenceID)":"3"}
}
```


backend

```
<?php
$ID=$_GET["id"];
include_once ("connect.php");

$same_refer=mysqli_query($conn,"SELECT a.PaperID,b.Title,b.PaperPublishYear,count(a.
    ReferenceID) FROM paper_reference a
                                    LEFT JOIN papers b ON a.PaperID=b.PaperID
```

```
                                                WHERE ((a.ReferenceID in (SELECT
                                                    ReferenceID FROM paper_reference WHERE
                                                     PaperID='$ID') and a.PaperID!='$ID')
                                                    or a.ReferenceID='$ID')
                                                GROUP BY a.PaperID ORDER BY count(a.
                                                    ReferenceID) DESC limit 7;");

$res=array();
while ($row=mysqli_fetch_assoc($same_refer))
{
    $res[]=json_encode($row);

}
echo json_encode($res);
?>
```

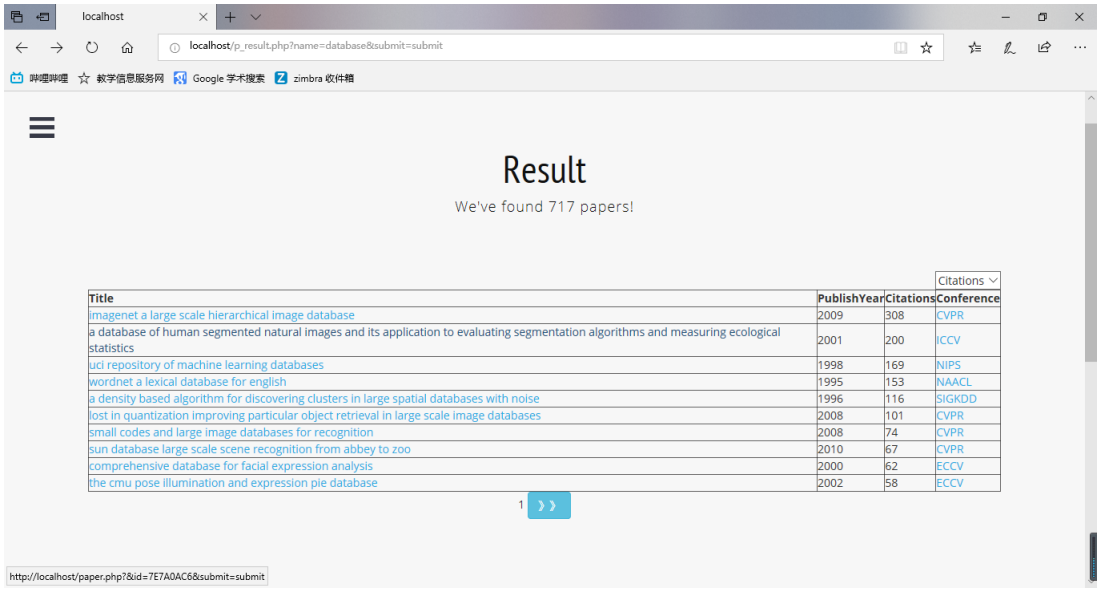
search page

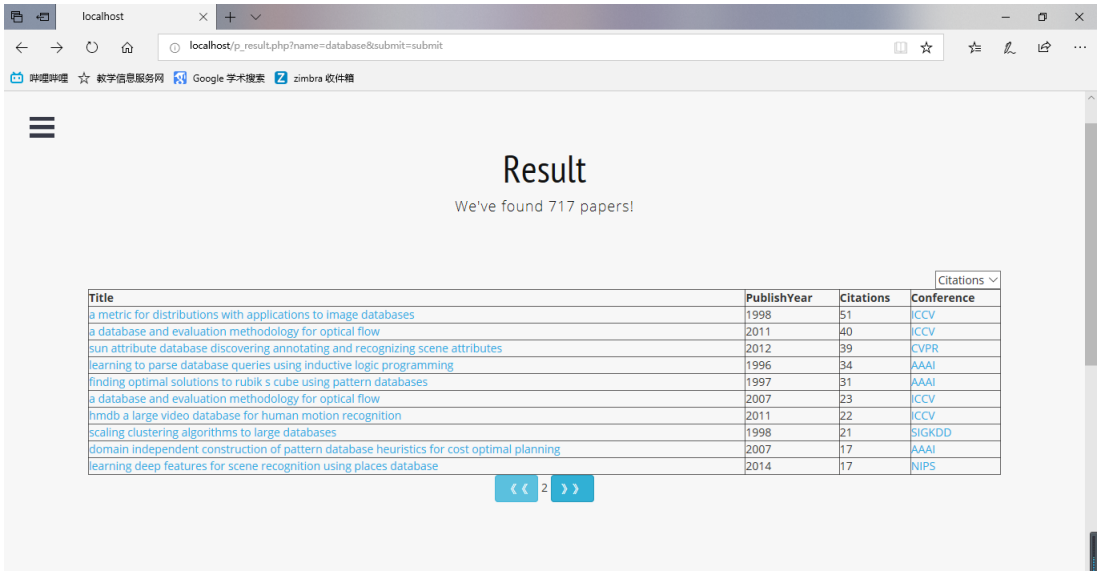
If the user needs to search another paper,he don't have to return the home page.For the users' convenience,in the end of the paper page,user can input another title and search it directly.

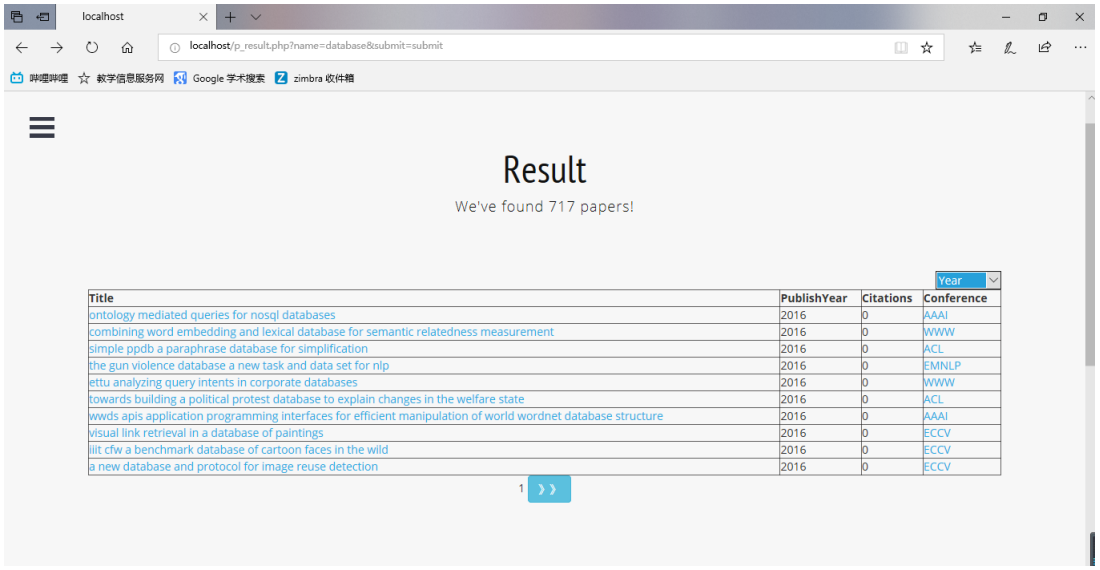
```
<section id="search" class="contact contact-section">
    <div class="container">
        <div class="row">
            <div class="col-lg-12">
                <div class="section-title text-center wow fadeInDown" data-wow-
                    duration="2s" data-wow-delay="50ms">
                    <h2>Search Paper</h2>
                    <p>What to know more?<br>Try to seaarch another paper!</p>
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col-lg-12">
                <center><form method="get" action="p_result.php">
                        <input type="text" placeholder="Author Name *" id="keysearch"
                            name="name" size="60"><br>
                        <button type="submit" name="submit" value="submit" class="btn
                            btn-primary waves-effect">Submit</button>
                    </form></center>
            </div>
        </div>
    </div>
</section>
```

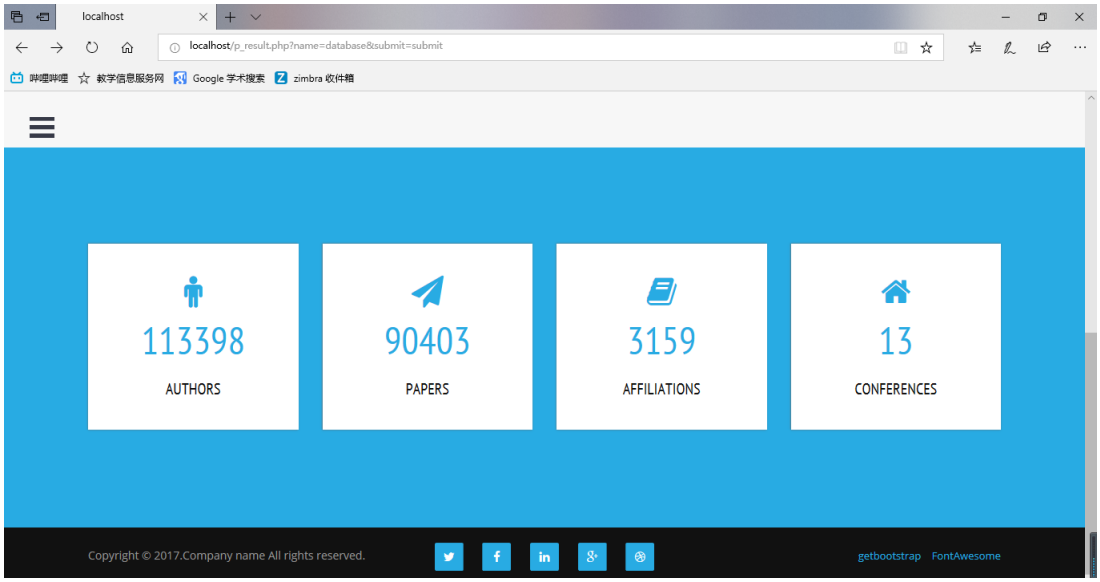
4.3 effect picture

p_result.php

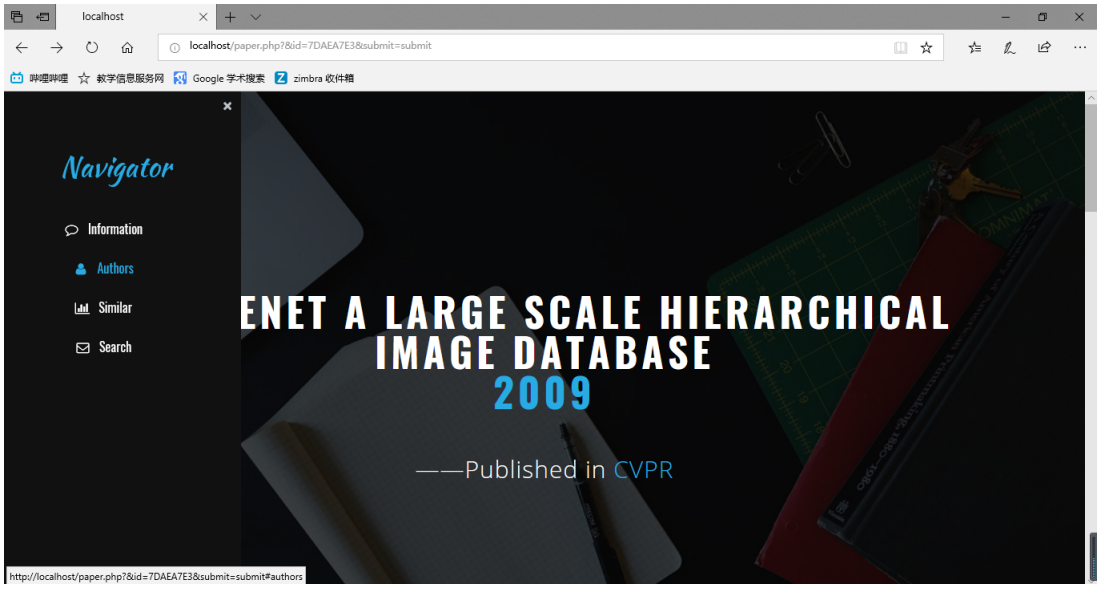

localhost
localhost/p_result.php?name=database&submit=submit
Result
We've found 717 papers!
Citations
Title PublishYear Citations Conference
imagenet a large scale hierarchical image database 2009 308 CVPR
a database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics 2001 200 ICCV
uci repository of machine learning databases 1998 169 NIPS
wordnet a lexical database for english 1995 153 NAACL
a density based algorithm for discovering clusters in large spatial databases with noise 1996 116 SIGKDD
lost in quantization improving particular object retrieval in large scale image databases 2008 101 CVPR
small codes and large image databases for recognition 2008 74 CVPR
sun database large scale scene recognition from abbey to zoo 2010 67 CVPR
comprehensive database for facial expression analysis 2000 62 ECCV
the cmu pose illumination and expression pie database 2002 58 ECCV
1
http://localhost/paper.php?&id=7E7A0AC6&submit=submit

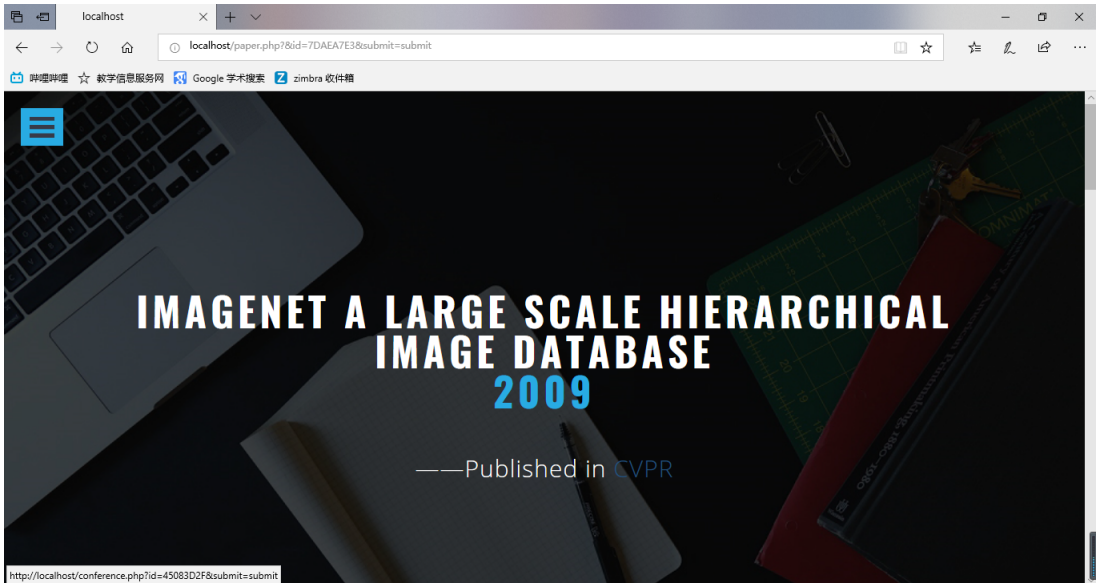

localhost
localhost/p_result.php?name=database&submit=submit
Result
We've found 717 papers!
Citations
Title PublishYear Citations Conference
a metric for distributions with applications to image databases 1998 51 ICCV
a database and evaluation methodology for optical flow 2011 40 ICCV
sun attribute database discovering annotating and recognizing scene attributes 2012 39 CVPR
learning to parse database queries using inductive logic programming 1996 34 AAAI
finding optimal solutions to rubik s cube using pattern databases 1997 31 AAAI
a database and evaluation methodology for optical flow 2007 23 ICCV
hmdb a large video database for human motion recognition 2011 22 ICCV
scaling clustering algorithms to large databases 1998 21 SIGKDD
domain independent construction of pattern database heuristics for cost optimal planning 2007 17 AAAI
learning deep features for scene recognition using places database 2014 17 NIPS
2

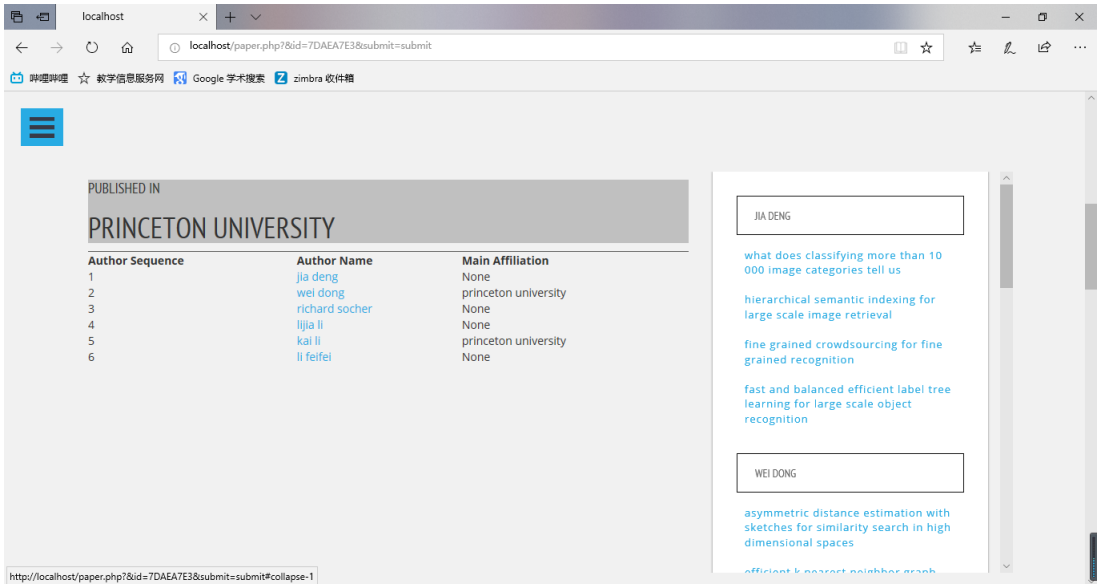

localhost
localhost/p_result.php?name=database&submit=submit
Result
We've found 717 papers!
Year
Title
PublishYear
Citations
Conference
ontology mediated queries for nosql databases
2016
0
AAAI
combining word embedding and lexical database for semantic relatedness measurement
2016
0
WWW
simple ppdb a paraphrase database for simplification
2016
0
ACL
the gun violence database a new task and data set for nlp
2016
0
EMNLP
ettu analyzing query intents in corporate databases
2016
0
WWW
towards building a political protest database to explain changes in the welfare state
2016
0
ACL
wwds apis application programming interfaces for efficient manipulation of world wordnet database structure
2016
0
AAAI
visual link retrieval in a database of paintings
2016
0
ECCV
iiit cfw a benchmark database of cartoon faces in the wild
2016
0
ECCV
a new database and protocol for image reuse detection
2016
0
ECCV
1

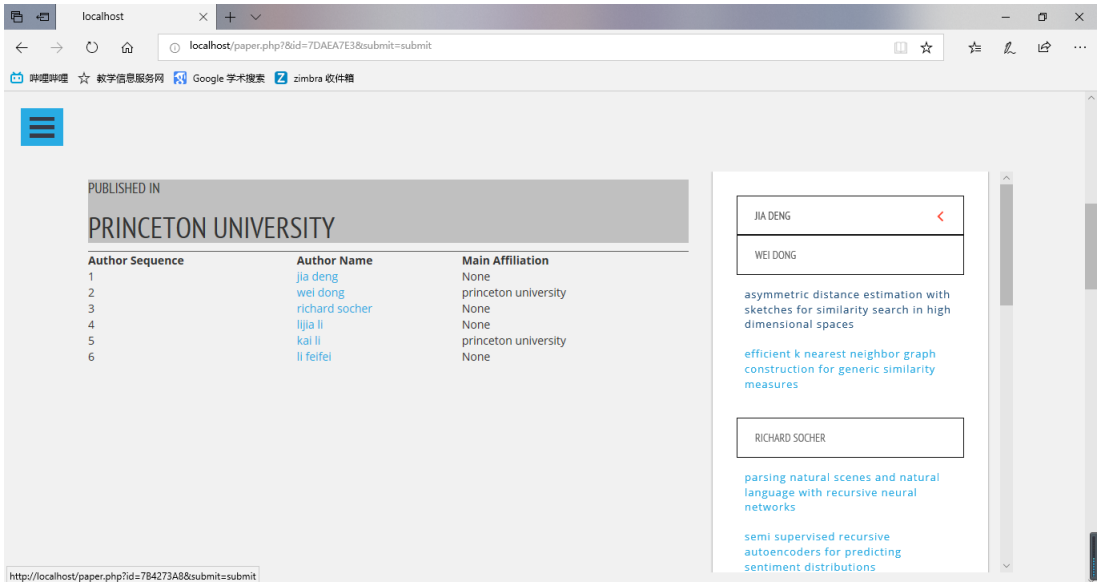

localhost
localhost/p_result.php?name=database&submit=submit
113398
AUTHORS
90403
PAPERS
3159
AFFILIATIONS
13
CONFERENCES
Copyright © 2017.Company name All rights reserved.
getbootstrap
FontAwesome

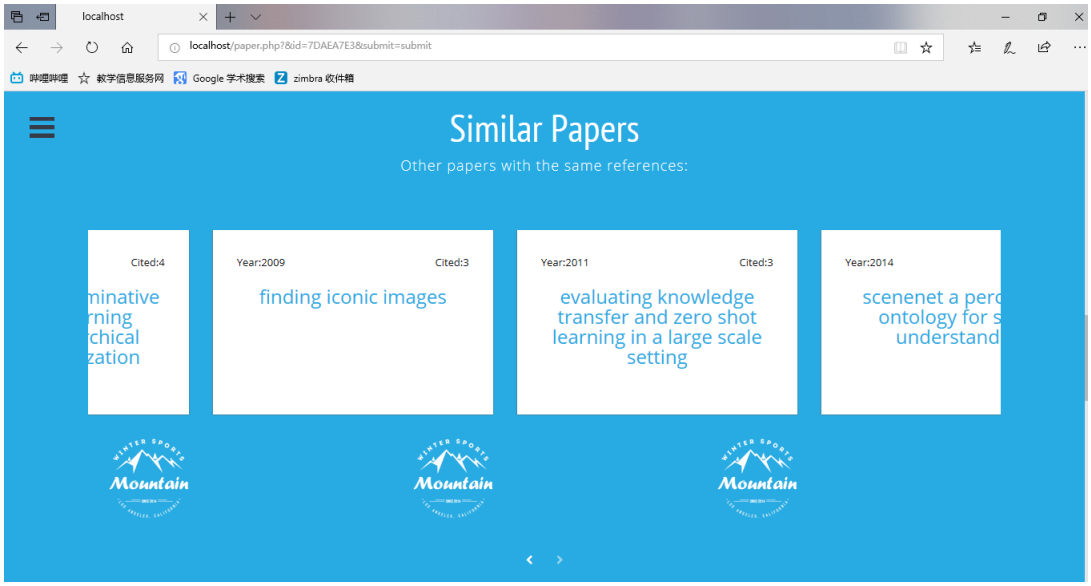
paper.php

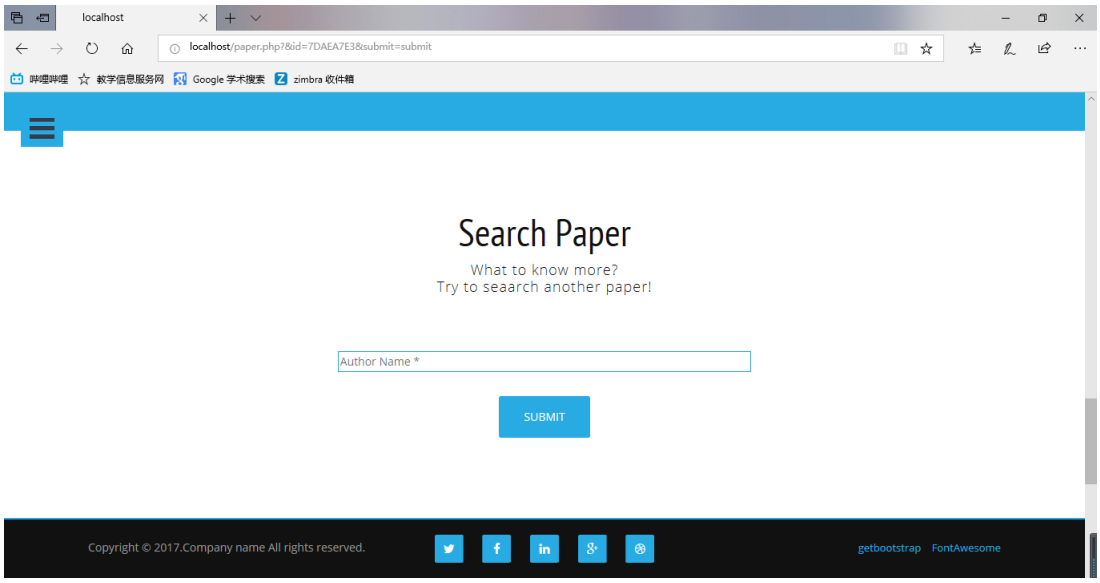

localhost
localhost/paper.php?&id=7DAEA7E3&submit=submit
Navigator
Information
Authors
Similar
Search
ENET A LARGE SCALE HIERARCHICAL
IMAGE DATABASE
2009
——Published in CVPR
http://localhost/paper.php?&id=7DAEA7E3&submit=submit#authors


localhost
localhost/paper.php?&id=7DAEA7E3&submit=submit
IMAGENET A LARGE SCALE HIERARCHICAL
IMAGE DATABASE
2009
——Published in CVPR
http://localhost/conference.php?id=45083D2F&submit=submit


localhost
localhost/paper.php?&id=7DAEA7E3&submit=submit
PUBLISHED IN
PRINCETON UNIVERSITY
Author Sequence
Author Name
Main Affiliation
1 jia deng None
2 wei dong princeton university
3 richard socher None
4 lijia li None
5 kai li princeton university
6 li feifei None
JIA DENG
what does classifying more than 10 000 image categories tell us
hierarchical semantic indexing for large scale image retrieval
fine grained crowdsourcing for fine grained recognition
fast and balanced efficient label tree learning for large scale object recognition
WEI DONG
asymmetric distance estimation with sketches for similarity search in high dimensional spaces
http://localhost/paper.php?&id=7DAEA7E3&submit=submit#collapse-1


localhost
localhost/paper.php?&id=7DAEA7E3&submit=submit
PUBLISHED IN
PRINCETON UNIVERSITY
Author Sequence
Author Name
Main Affiliation
1 jia deng None
2 wei dong princeton university
3 richard socher None
4 lijia li None
5 kai li princeton university
6 li feifei None
JIA DENG
WEI DONG
asymmetric distance estimation with sketches for similarity search in high dimensional spaces
efficient k nearest neighbor graph construction for generic similarity measures
RICHARD SOCHER
parsing natural scenes and natural language with recursive neural networks
semi supervised recursive autoencoders for predicting sentiment distributions
http://localhost/paper.php?id=7B4273A8&submit=submit





5 Visualization and Another Search BOX (Yuting Lan)

5.1 Visualization of Force-Directed Graph

5.1.1 Brief Introduction of Force-Directed Graph in our project

In each author page, we decided to add a visual map of this scholar and all his collaborators. That is to find all the partners of author, and find all the cooperative relationships among these scholars, then visualize the network structure between these points. We can use

the Force-Directed Graph framework in reference link 1. At the same time, the point of the current scholar and his collaborator should be distinguished from different colors.

5.1.2 Problem Analysis and Key Codes

```
var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height");

var color = d3.scaleOrdinal(d3.schemeCategory20);

var simulation = d3.forceSimulation()
    .force("link", d3.forceLink().id(function(d) { return d.id; }))
    .force("charge", d3.forceManyBody())
    .force("center", d3.forceCenter(width / 2, height / 2));
```

The 'width' and the 'height' is the parameter about the width and height of the whole environment. The 'color' is about the color type of the points. The 'simulation' sets the parameters of the simulation model.

```
d3.json("test.php?authorname=$name", function(error, graph) {
  if (error) throw error;

  var link = svg.append("g")
      .attr("class", "links")
    .selectAll("line")
    .data(graph.links)
    .enter().append("line")
      .attr("stroke-width", function(d) { return Math.sqrt(d.value)
                                          ; });

  var node = svg.append("g")
      .attr("class", "nodes")
    .selectAll("circle")
    .data(graph.nodes)
    .enter().append("circle")
      .attr("r", 5)
      .attr("fill", function(d) { return color(d.group); })
      .call(d3.drag()
          .on("start", dragstarted)
          .on("drag", dragged)
          .on("end", dragended));

  node.append("title")
      .text(function(d) { return d.id; });

  simulation
      .nodes(graph.nodes)
      .on("tick", ticked);
```

```
30      simulation.force("link")
            .links(graph.links);
32
      function ticked() {
34      link
            .attr("x1", function(d) { return d.source.x; })
36          .attr("y1", function(d) { return d.source.y; })
            .attr("x2", function(d) { return d.target.x; })
38          .attr("y2", function(d) { return d.target.y; });

40      node
            .attr("cx", function(d) { return d.x; })
42          .attr("cy", function(d) { return d.y; });
      }
44  });
```

I use test.php to store the imformation which will be used to draw the forece-directed graph,the other part just the same as the reference data.

```
    function f1(d) {
2     if (!d3.event.active) simulation.alphaTarget(0.3).restart();
      d.fx = d.x;
4     d.fy = d.y;
    }
6
    function f2(d) {
8     d.fx = d3.event.x;
      d.fy = d3.event.y;
10  }

12  function f3(d) {
      if (!d3.event.active) simulation.alphaTarget(0);
14    d.fx = null;
      d.fy = null;
16  }
```

I use function f1,f2,f3 to exhibit the drag process.

5.1.3 Result Exhibition

Figure 1: display of force-directed-graph

5.2 Visualization of Tree graph

5.2.1 Feature Extraction

As lab3,We should Implement a function whose parameters are any two scholar IDs that have been collaborating and can extract the characteristics of the cooperative relationship between the two scholars.we are provide 9 different charcteristics about A and B. In the later exercise ,we will use it to train our models later.

```
authorid_authorname = []
with open('data/authors.txt', 'r', encoding = 'utf-8') as file:
    counter = 0
    while True:
        content = file.readline()
        # content = content.decode('utf-8')
        if content == "":
            break
        if content =='\n':
            continue
        counter += 1
        details = content.split('\t')
        authorid_authorname.append([details[0], details[1]])

print("authorid_authorname: finished\n")


paperid_title_year = []
with open('data/papers.txt', 'r', encoding = 'utf-8') as file:
    counter = 0
    while True:
        content = file.readline()
        # content = content.decode('utf-8')
        if content == "":
```

```
            break
        if content ==’\n’:
            continue
        counter += 1
        details = content.split(’\t’)
        paperid_title_year.append([details[0], details[1], details[2]])

print("paperid_title_year: finished\n")


paper_author_affiliation = []
with open(’data/paper_author_affiliation.txt’, ’r’, encoding = ’utf-8’)
                                        as file:
    counter = 0
    while True:
        content = file.readline()
        # content = content.decode(’utf-8’)
        if content == "":
            break
        if content ==’\n’:
            continue
        counter += 1
        details = content.split(’\t’)
        paper_author_affiliation.append([details[0], details[1],
                                                 details[2]])

print("paper_author_affiliation: finished\n")
```

To improve the speed , I didn’t use the code of lab3,I choose use the txt directly,rather than sql sentences .After using the above python code ,I extract every information with high speed.

```
# this function search the infomation of a specific author
# parameters: authorid
# return: [paperid, authorid, affiliationid, publishyear]
def allPaper(id):
    idpaper = []
    for item1 in paper_author_affiliation:
        if item1[1] == id:
            for item2 in paperid_title_year:
                    if item2[0] == item1[0]:
                        idpaper.append([item1[0], item1[1], item1[2],
                                                            item2[2]])
                        break
    return idpaper

# this function finds the year of an author’s first paper
# parameters: [paperid, authorid, affiliationid, publishyear]
# eturn: year
def firstYear(paperlist, para = 3):
    year = 3000
    for item in paperlist:
```

```
        year = min(int(item[para]), year)
    return year

# this function finds the year of an author's last paper
# parameters: [paperid, authorid, affiliationid, publishyear]
# return: year
def lastYear(paperlist, para = 3):
    year = 0
    for item in paperlist:
        year = max(int(item[para]), year)
    return year

# this function finds how many papers an author had published before
                                    the cooperation
# parameters: [paperid, authorid, affiliationid, publishyear], yaer
# return: beforeyear
def beforePaper(paperlist, year):
    beforepaper = 0
    for item in paperlist:
        if int(item[para]) < year:
            beforepaper += 1
    return beforepaper

#this function finds the cooperation of two authors
# parameters: [paperid, authorid, affiliationid, publishyear], [paperid
                                    , authorid, affiliationid,
                                    publishyear]
#return: [paperid, authorid, affiliationid, publishyear]
def sharedPaper(paperlist1, paperlist2):
    sharedpaper = []
    for item1 in paperlist1:
        for item2 in paperlist2:
            if item1[0] == item2[0]:
                sharedpaper.append([item1[0], item1[1], item2[1], item1
                                                        [3]])
    return sharedpaper

#this function finds out how many papers A published during their
                                    cooperation
# parameters: [paperid, authorid, affiliationid, publishyear], [paperid
                                    , authorid, affiliationid,
                                    publishyear], firstyear, lastyear
# return: withoutb
def aWithNoB(allpaper, sharedpaperlist, firstyear, lastyear):
    withoutb = 0
    for item1 in allpaper:
        if int(item1[3]) >= firstyear and int(item1[3]) <= lastyear:
            isdif = 0
            for item2 in sharedpaperlist:
                if item1[0] == item2[0]:
                    isdif = 1
                    break
```

```
            if isdif == 0:
                withoutb +=1
    return withoutb

# this function prints the paperlist
# parameters: [paperid, authorid, affiliationid, publishyear]
#return: null
def printlist(printlist):
    for item in printlist:
        print(item)
    print("\n")

#this function finds out how many papers A published during their cooperation
# parameters: id1, id2
# return: list of features
def extract(id1, id2):
    id1paper = allPaper(id1)
    id2paper = allPaper(id2)
    sharedpaperlist = sharedPaper(id1paper, id2paper)
    firstyear = firstYear(sharedpaperlist)
    firstyear1 = firstYear(id1paper)
    firstyear2 = firstYear(id2paper)
    lastyear = lastYear(sharedpaperlist)
    lastyear1 = lastYear(id1paper)
    lastyear2 = lastYear(id2paper)

    sharedpaper = len(sharedpaperlist)
    beforepaper1 = beforePaper(id1paper, firstyear)
    beforepaper2 = beforePaper(id2paper, firstyear)
    awithnob = aWithNoB(id1paper, sharedpaperlist, firstyear, lastyear)
    bwithnoa = aWithNoB(id2paper, sharedpaperlist, firstyear, lastyear)

    feature = []
    feature.append(beforepaper1)
    feature.append(beforepaper2)
    feature.append((beforepaper1 - beforepaper2) / sharedpaper)
    feature.append(firstyear2 - firstyear1)
    feature.append(firstyear - firstyear2)
    feature.append((firstyear2 - firstyear1 - firstyear + firstyear2) / (lastyear - firstyear + 1))
    feature.append(awithnob)
    feature.append(bwithnoa)
    feature.append((awithnob - bwithnoa) / sharedpaper)

    return feature
```

The use of the function are showed in the code Then I use tensorflow to train and test it.Store it in the master.txt

```
myfile=open("fearture_extraction.txt","r")
```

```
L1=[]
while True:
    theline=myfile.readline()
    theline2=theline.rstrip("\n")
    if len(theline2)==0:
        break
    theline3=eval(theline2)
    for i in theline3:
        i=float(i)
    L1.append(theline3)

A1=np.array(L1)
```

In the course of dealing with data ,.In the first place ,we should fetch our characteristics in the file .After that ,we should change the type of data into float ,because the data type we will use in the tensorflow is float . so we use for sentences to transform every item in the list to float type .

```
myfile1=open("train.txt","r")
L2=[]
while True:
    theline=myfile1.readline()
    theline2=theline.rstrip("\n")

    if len(theline2)==0:
        break

    theline3=theline2.split("\t")
    if theline3[2]=="1":
        L2.append([0.,1.])
    else:
        L2.append([1.,0.])
A2=np.array(L2)
```

In addition ,we also need to transform label type . type must be float . 1 dimentionnal array must change into type 2 dimentional array . for instance ,I change label 1 for [0.,1.] .Then I change lable 0 for [1.,0.]

//remark: the data transform of test is just the same .So i don't state it repeatedly .

```
import tensorflow as tf
sess = tf.InteractiveSession()

x = tf.placeholder(tf.float32, [None, 9])
W = tf.Variable(tf.zeros([9, 2]))
b = tf.Variable(tf.zeros([2]))
```

The first step is to define algorithm formula to build a placeholder, None means that the number of samples can be arbitrary, 9 is for we have our 9 features for every author .2 means our label is two dimention

```
y = tf.nn.sigmoid(tf.matmul(x, W) + b)
y = tf.nn.softmax(tf.matmul(x, W) + b)
y = tf.nn.tanh(tf.matmul(x, W) + b)
y = tf.nn.relu(tf.matmul(x, W) + b)
```

Then we set up a model ,I have try on the different activation function ,such as softmax,tanh ,sigmoid,relu . activation function means differet math formula .Obviousely some of the activation is not consistent with our data so that the accuracy is too low .

```
\\Normalization
\\use sklearn
from sklearn import preprocessing
min_max_scaler=preprocessing.MinMaxScaler()
A1=min_max_scaler.fit_transform(A1)

A3=min_max_scaler.fit_transform(A3)
```

Normalization can speed up the speed of our project ,I choose the package sklearn like before .

```
cross_entropy = -tf.reduce_sum(y_ * tf.log(y))

train_step = tf.train.GradientDescentOptimizer(0.01).minimize(
                                        cross_entropy)

tf.global_variables_initializer().run()

train_step.run({x: A1, y_: A2})
```

GradientDescent function can minimize cross_entropy loss function using gradient descent . We name it as train step ,then we use it to train the model with train_feature .

```
with open('master.txt', 'w', encoding = 'utf-8') as writein:
    with open('allcooperation.txt', 'r', encoding = 'utf-8') as file:
        counter = 0
        init = time.time()
        while True:
            # if counter == total:
            #   break
            content = file.readline().replace("\n", "")
            # content = content.decode('utf-8')
            if content == "":
                break
            if content =='\n':
```

```
                continue
            counter += 1
            details = content.split('\t')
            # train_X.append([details[0], details[1], int(details[2])])
            # view_bar(counter, total)
            if counter % 100 == 0:
                print(counter, end = '')
                print(":  ", end = '')
                print((300000 / counter - 1) * (time.time() - init))
            # print(details[0], details[1])
            featurelist = extract(details[0], details[1])
            for i in range(9):
                featurelist[i] = float(featurelist[i])
            # predict_X.append(np.array(featurelist))
            tag = sess.run(y, feed_dict={x: np.array([featurelist])})
            # print(tag[0])
            flag = tag[0][0] > tag[0][1]
            writein.write(details[0] + "\t" + details[1] + "\t" + str(
                                                flag) + "\n")
            writein.write(details[1] + "\t" + details[0] + "\t" + str(
                                                not flag) + "\n")
            # print(featurelist)
            # feature.write(details[0] + "\t" + details[1] + "\t")
            # for item in featurelist:
            #     feature.write(str(item) + "\t")
            # feature.write("\n")
            # feature.write(str(details[2]) + "\n")
# predict_X = np.array(predict_X)


# predict_X = preprocessing.MinMaxScaler().fit_transform(predict_X)
# predict_y = preprocessing.MinMaxScaler().fit_transform(predict_y)
```

Then we use the model of lab3 ,to test our data .

5.2.2 Database Establishment

```
cursor.execute("""CREATE TABLE Predictions(
  Master char(8),
  Student char(8)
  )
  """)
```

For the convenience of search the data,I decided to establish a database to store the teacher - student information .This table(predictions) has two columns ,the first one which is named after Master,the second one is named after Student .I store the cooperation schlors ID in this table.

```
with open('master.txt', 'r', encoding = 'utf-8') as paper:
  counter = 0
  while True:
    # try:
    content = paper.readline()
    # content = content.decode('utf-8')
    if content == "":
      break
    counter += 1
    print(counter, ": ", content)
    details = content.split('\t')
    # print(details[2])
    if(details[2] == "True\n"):
      cursor.execute("""INSERT INTO Predictions VALUES ('{}','{}')""".
                                    format(details[0], details[1]))
    # break
  connection.commit()

    # except:
    #   cursor.execute("""INSERT INTO Papers VALUES ('{}','{}','Error
                                    ','{}','{}')""".format(str(counter)
                                    , details[0], details[2], details[3
                                    ]))
    #   connection.commit()
    #   continue
```

I open the txt file master.txt which we store the data in the file .So we use python to deal the data, use INSERT sentences to INSERT it to the database .To inprove our dafabase table , I decided to just store the test results for True's teacher-student relationship. So ,the speed of serach will be up.Then i add two index .

5.2.3 The Implement of Back End

```
$dbhost = 'localhost:3306';
$dbuser = 'root';
$dbpass = '';
$conn = mysqli_connect($dbhost, $dbuser, $dbpass);

if(! $conn )
{
    die('fail: ' . mysqli_error($conn));
}
else
{
  mysqli_query($conn , "set names utf8");

  mysqli_select_db( $conn, 'ieee2018lab1' );
```

The same as the formal project ,we firstly connected our database ,which has the prediction table .

```
$retval = mysqli_query( $conn,
    "SELECT
      AuthorName
    From authors a
          INNER JOIN
      (SELECT Master as MasterID,Student as StudentID
        From predictions a
      INNER JOIN
        (SELECT AuthorID as ID From authors
                    WHERE AuthorName= '$name' ) b
      ON a.Master=b.ID) b
    ON a.AuthorID=b.StudentID");
```

Then ,I make a brief introduction for our sql sentences .To better speed up the inquiry, I use the Nesting of SQL statements . The first layer is that we use the AuthorName ,find the related AuthorID in the table authors . The second layer is that we the the AuthorID as Masters ID ,find the related StudentID in the table Predictions . The Third layer is that we use the StudentID as AuthorID ,find the related AuthorName.

```
$retval2 = mysqli_query( $conn,
    "SELECT
      AuthorName
    From authors a
          INNER JOIN
      (SELECT Master as MasterID,Student as StudentID
        From predictions a
      INNER JOIN
        (SELECT AuthorID as ID From authors
                    WHERE AuthorName= '$name' ) b
      ON a.Student=b.ID) b
    ON a.AuthorID=b.MasterID");
```

This sentences just change a little bit ,The second layer is that we the the AuthorID as StudentID ,find the related MasterID in the table Predictions

```
while($row = mysqli_fetch_array($retval, MYSQL_ASSOC))
{
  $result[]=json_encode($row);

}
$result=json_encode($result);
echo $result;
```

Then ,seach the imformation and store it in the json array .We use echo to transmit it .

5.2.4 The Implement of Front End

I show the fundamental Idea of Front End,Firstly ,we should get the json information from back end .In addition ,we should use the information to draw a tree . How to implement it ,I seach the internet and choose d3.js.tree .

```
<style>

  .node {
    cursor: pointer;
  }

  .node circle {
    fill: #fff;
    stroke: steelblue;
    stroke-width: 1.5px;
  }

  .node text {
    font: 10px sans-serif;
  }

  .link {
    fill: none;
    stroke: #ccc;
    stroke-width: 1.5px;
  }
  .tree{
    width: 800px;
    height: 800px;
    margin: 0 auto;
    background: #E0E0E0;
  }
  .tree svg{
      width: 100%;
      height: 100%;
  }

</style>
```

This is some fundamental setting of our tree . We set the color of the link ,node cicle ,node text . In addition ,we set the width and length of the tree,node.

```

content="[{"+"\"name\":"+jsname+",\"parent\":\"null\","+"\"children\":
                                        "";
```

```
function write_array(namei)
{
  $.getJSON("tree.php",{name:namei},function(result){

  if (result.length==0)
    content+="[]";
  else
  {
    for (var i=0;i<result.length;++i)
    {
      item=eval("("+result[i]+")");
      //alert(item['AuthorName']);
      content+="[{"+"\"name\":"+item['AuthorName']+",\"parent\":"+namei
                                    +",+\"children\":";
      //write_array(item['AuthorName'],jsoni);
      write_array(item['AuthorName']);
      content+="}]";
    }
  }

  content+="}]";

  content=(eval("("+content+")"))

      })

}
```

The most important part is two produce a json array ,so I define a function write_array().In this function ,firstly we use getJson to get the json data from the back end ,like [{"AuthorName":asda}.....], then we define a string content ,use it to form the array we need .As content+=.... .After that we should set the end condition.If the length of result is equal to 0. To find the children of parent ,we need to call write_array again in the for sentences.

```
  var margin = [20, 120, 20, 120],
      width = document.getElementById("tree").offsetWidth,
      height = document.getElementById("tree").offsetHeight;

  var i = 0,
      duration = 750,
      root;

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

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

  var zoom = d3.behavior.zoom().scaleExtent([0.1, 100]).on("zoom",
                                    zoomed);
```

```
var svg = d3.select("body").select("#tree").append("svg")
.call(zoom)
  .append("g")
  .call(zoom)//
  .append("g")
    .attr("transform", "translate(" + margin[3] + "," + margin[0] + "
                                    )");
```

In the above sentences, we give some settings . In addition ,Add magnification and shrink events.That means ,if we press the node ,this node and it's children will become larger . It's provide the better feeling for the users .We also Binding zoom events to G and Binding zoom events to svg.

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

    nodeEnter.append("circle")
        .attr("r", 1e-6)
        .style("fill", function(d) { return d._children ? "
                                    lightsteelblue" : "#fff"; });

    nodeEnter.append("text")
        .attr("x", function(d) { return d.children || d._children ? -10
                                     : 10; })
        .attr("dy", ".35em")
        .attr("text-anchor", function(d) { return d.children || d.
                                    _children ? "end" : "start"; })
        .text(function(d) { return d.name; })
        .style("fill-opacity", 1e-6);
```

This is about the setting about node.enter .That means that ,

If we press the node, it will extend many branches.

```
function click(d) {
  if (d.children) {
    d._children = d.children;
    d.children = null;
  } else {
    d.children = d._children;
    d._children = null;
  }
  update(d);
}
```

This is the click function ,That means , if this node has childre node ,it will extend. else stop.

5.2.5 Result Exhibition

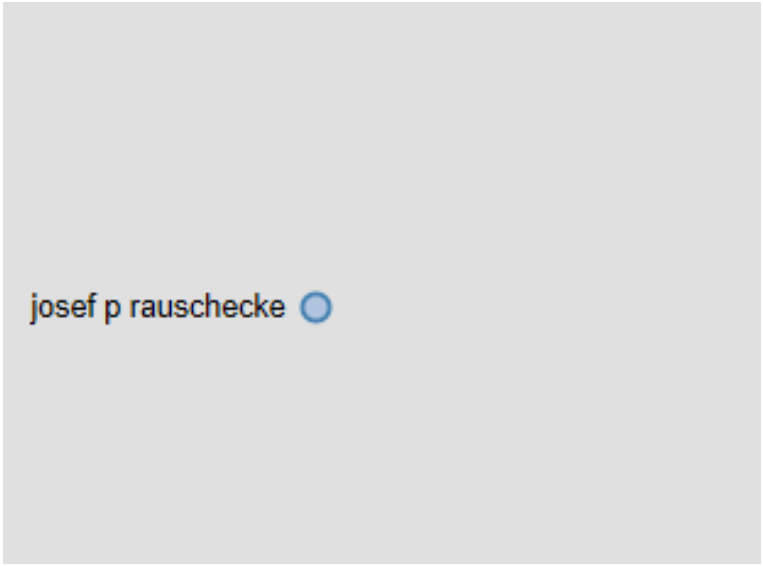


Figure 2: display of Tree graph

If you press the button, it will extend and automatically become larger

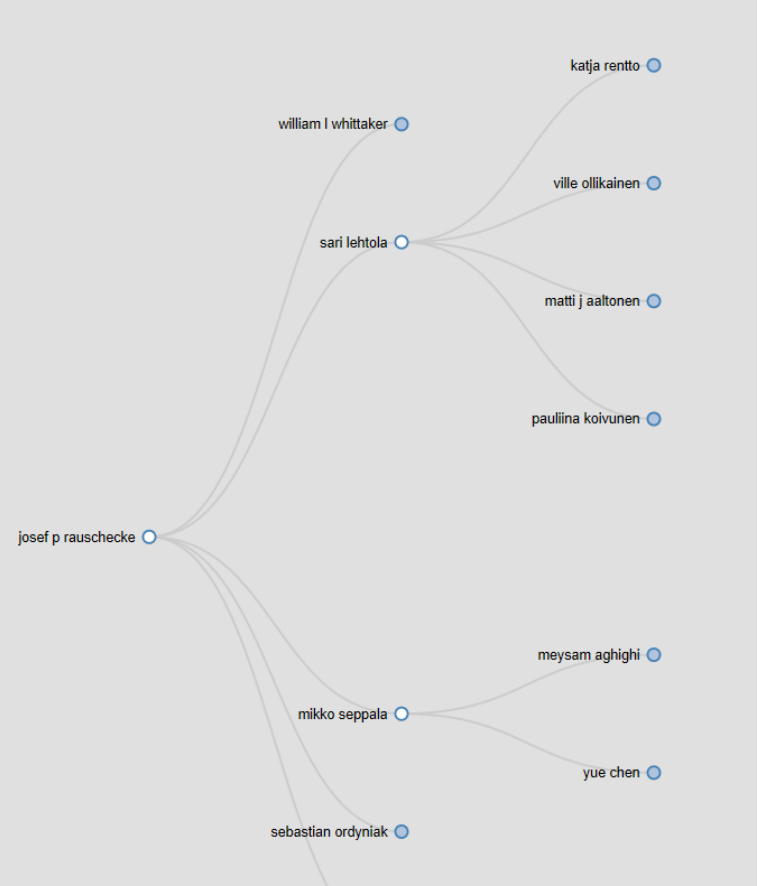


Figure 3: display of Tree graph

Just like this.

5.3 Another Search Box

5.3.1 Brief Introduction

Besides the related search box,there is another type of search box .

5.3.2 The Implement of Another Seach Box

```
<div class="searchbox">
  <ul class="border1">

    <li><a href='#'>Paper Search</a></li>
    <li><a href="#">Conference Search</a></li>
    <li><a href="#">Author Search</a></li>
  </ul>
  <div class="bodys">

```

```
    <p><input type="text" value="" id=""  name="name" class="one"
                                    placeholder="Please input
                                    AuthorName" /><button class="one1">
                                    Search</button></p>

    <p><input type="text" value="" id="" class="two" placeholder="
                                    Please input ConferenceName" /><
                                    button class="two2">Search</button>
                                    </p>
    <p><input type="text" value="" id="" class="three" placeholder="
                                    Please input PaperName" /><button
                                    class="three3">Search</button></p>
  </div>
</div>
```

In this section, I create three buttons .These three buttons related to different search ,Author,Paper,Reference. I set the placeholder as different search hint and give them different class .

```
<script>
$(function(){
  $(".bodys p").not(":first").hide();
  $(".searchbox ul li").mouseover(function(){
    var index = $(this).index();
    if(index==0){
      $(this).find("a").addClass("style1");
      $("li").eq(1).find("a").removeClass("style2");
      $("li").eq(2).find("a").removeClass("style3");
    }
    if(index==1){
      $(this).find("a").addClass("style2");
      $("li").eq(0).find("a").removeClass("style1");
      $("li").eq(2).find("a").removeClass("style3");
    }
    if(index==2){
      $(this).find("a").addClass("style3");
      $("li").eq(0).find("a").removeClass("style1");
      $("li").eq(1).find("a").removeClass("style2");
    }
    var index=$(this).index();
    $(".bodys p").eq(index).show().siblings().hide();
  });
});
</script>
```

Then ,we create a function ,use element .Set a a mouseover function .That means that,if you put your mouse on it , In would change box,and hide this box.

```
<style type="text/css">
*{margin:0;padding:0;list-style-type:none;}
a,img{border:0;}
/* searchbox */
.searchbox{width:520px;height:80px;margin:40px auto 0 auto;}
.searchbox ul{ height:35px; width:500px; list-style:none; margin-left:
                                    20px}
.searchbox ul li{ float:left}
.searchbox ul li a{ float:left; line-height:35px; padding:0 20px; text-
                                    decoration:none; color:#000; font-
                                    size:14px; font-weight:bold;}
.searchbox ul li .style1{ background-color:#000; color:#fff}
.searchbox ul li .style2{ background-color:#f00;color:#fff}
.searchbox ul li .style3{ background-color:#F90;color:#fff}


.bodys input{ height:30px;line-height:30px;width:390px;padding:0 10px;
                                    float:left;}
.bodys .one{ border:#99BBFF 3px solid}
.bodys .two{ border:#0066FF  3px solid}
.bodys .three{ border:#33FFFF 3px solid}
.bodys .one1{ background-color:#99BBFF; }
.bodys .two2{ background-color:#0066FF;}
.bodys .three3{ background-color:#33FFFF;}
.bodys button{float:left;  border:0;  height:36px;  width:100px; color:
                                    #0044BB; line-height:36px;text-
                                    align:center;overflow:hidden;}
</style>
```

This is the color set ,and size (width, height set.

5.3.3 Result Exhibition

Figure 4: display of Box

6 Optimizing (Dongwei Xie)

My job is to debug the code, do beautification, and help construct the home page. To do beautification, I need to design the layout and the codes. To design the layout, the principle is to make the website more user-friendly and beautiful. So I add three optional button which

Figure 5: display of Box

represents Author, paper and conference. To speed up, we need to optimize the codes, the principle is as following: 1. Use “echo” instead of “print”. The efficiency of “echo” is higher than that of “print”, because “echo” has no return value, yet “print” returns an integer. 2. Destroy variables to release memory, especially large arrays. 3. use PHP’s internal string to manipulate functions, try to use them instead of using regular expressions. 4. Use single quotation marks instead of double quotes to contain strings, which will be quicker. 5. Don’t copy variables casually. Sometimes to make the PHP code clean, we will copy the predefined variables into a simpler variable with a shorter name, and in fact, the result is a doubling of memory consumption and slows the program. Based on the principles above, I optimize the codes, and the result seems positive.

7 PUTTING THE WEBSITE ONLINE (ZIHAN XU)

The website should be uploaded onto a real server to be accessed by anyone. Now I will show the concrete steps to put our project online.

(1) Get a host

Aliyun released a special count for students, for only 9.9 yuan per month, with 1G memory and 500G cloud storage. Since this part is not totally related to technical event, I will skip the procedures.

(2) Get a domain name

Domain names are what identify your site on the internet, such as www.baidu.com or www.google.com. Without such an address, you can only access your website through the IP Address of the server on which it is hosted. In English, this means that instead of having www.google.com as an address, you would end up with 216.239.51.99. This looks formidably unprofessional and it’s hard to remember.

But actually Ali also offered such service, which means we can buy a domain name directly.

(3) Link the domain to your host

Now, you need to connect the domain to the webhost, so that when you type the domain on the internet you are redirected to the site. How to do this depends on if you’re self-hosting or using a company’s server.

Since residential internet connections have dynamic IP addresses, that is, an IP that changes over time, it is impossible to simply link your IP to your URL. Fortunately, there is a program called DynDNS that can update the URL constantly, as the IP changes. Don’t remember that, for this to work though, you need to buy the URL from DynDNS or use a free

subdomain.

(4) Install Ubuntu Server on the cloud server

Actually Ali had already done this for me, so I only needed to connect to it.

(5) Connect with the cloud server and building LAMP on it

I used PuTTY to make connection with the could server:

(6) Building LAMP on it

A LAMP server comes with the complete package. Not only will you have the server, but you'll have the operating system, database software, and scripting language. All of these applications are open source. LAMP server encompasses the following components: Linux, Apache, MySQL, and PHP. LAMP servers can run on inexpensive servers. Constructing this server is simple and straightforward. Build a LAMP server by performing the following steps:

install apache2:

```
sudo apt-get install apache2
```

install PHP:

```
sudo apt-get install php5 libapache2-mod-php5
```

install Mysql:

```
sudo apt-get install mysql-server
```

The rest is easy as a piece of cake, and I will not waste time on it.

(7) Uploading whole website:

I used FileZilla to do the transportation.

Just write a script for FileZila:

```
<?xml version="1.0" encoding="UTF-8"?>
<FileZilla3 version="3.15.0.2" platform="*nix">
    <Servers>
        <Server>
            <Host>39.105.27.108</Host>
            <Port>22</Port>
            <Protocol>1</Protocol>
```

```
            <Type>0</Type>
            <User>root</User>
            <Pass encoding="base64">WHpoOTgxMDAx</Pass>
            <Logontype>1</Logontype>
            <TimezoneOffset>0</TimezoneOffset>
            <PasvMode>MODE_DEFAULT</PasvMode>
            <MaximumMultipleConnections>1</MaximumMultipleConnections>
            <EncodingType>Auto</EncodingType>
            <BypassProxy>0</BypassProxy>
            <Name>alicloud</Name>
            <Comments />
            <LocalDir />
            <RemoteDir />
            <SyncBrowsing>0</SyncBrowsing>
            <DirectoryComparison>0</DirectoryComparison>alicloud</Server>
    </Servers>
</FileZilla3>
```

With it we can have direct access to file system of the cloud server.

So my server address is: http://39.105.27.108/html
But I am sorry that due to limit of time, I could not be able to update the website online. What's on the web site is only a demo for this single part.

8 Conclusion (Dongwei Xie)

Project summary: By learning this course, we learned how to use the MySql database, how to make a website's front end and back end, how to use machine learning to extract features and how to do visualization. It is always of great benefit to learn such knowledge this semester.