北京大学信息科学技术学院

微信转发路径的获取与分析

网络实习实验报告

高嗣昂 孙周易 朱兆成 庄泽浩

2017-6-5

目录

1. 背景意义

1.1. 已有工作

在当今的的中国，人们用着不同品牌、不同操作系统的手机，但手机上会装着同样一个软件——微信。像曾经的 QQ、人人一样，微信已经成为了人们生活中不可或缺的一部分，人们在上面聊天、谈生意，在朋友圈分享自己的生活点滴。

正是由于微信是如此重要，有许多的企业希望通过微信推广自己的产品服务，因此微信 H5 页面的专业制作平台应运而生，如兔展 [1]、人人秀 [2]、易企秀 [3] 等。这些平台提供了大量的 H5 模版，让用户可以在短时间内生成精美的 H5 页面，并且在后台记录了关于这个页面被访问的详细信息，如浏览次数、阅读人数、转发次数、用户资料等，让用户知道传播效果如何。

1.2. 存在问题

虽然现阶段已有的平台已经足够易用，但还是存在着两大问题。一方面，虽然这些平台可以提供一定的统计数据如浏览次数等，但毕竟不是原始数据，而是经过了“加工”之后的结果，这就导致了原本可能有价值信息的丢失，例如转发路径等在后台就不一定提供；另一方面，这些平台毕竟是商用平台，如果想使用高级功能，例如查看详细数据等，是收费的，有的甚至达到 1500 元 / 月。但事实上这些平台也只是调用了微信的接口而已，而这些接口本身是免费的。可以说它们收费能做到的，我们自己不交钱也可以做到。

1.3. 技术挑战

第一个技术挑战就是要学习微信的“游戏规则”，了解如何获取到基本的用户信息、如何捕捉用户点击页面、转发的行为并加以记录。这需要阅读微信的相关开发者文档并在代码中实现。

第二个技术挑战是要编写能抓住用户眼球的 H5 页面。想要从转发路径中分析出有意义的信息，需要足够大的用户访问量，而这是要以内容足够优质、足够吸引人为前提的。因此我们有专门的编辑来编写内容，并在老师的指导下决定用翻页式 H5 来呈现内容。

第三个技术挑战是分析获取到的数据。我们最后获得了大量的数据，如何整理、分析这些数据得出有意义的信息，也是需要自己考量的方面。我们争取从转发路径中得出一些普适的、有价值的信息。

1.4. 工作意义

首先，这次实验需要我们学习并掌握了 python、JavaScript、HTML5、PHP 等语言，并利用这些编程语言合作完成我们的实验，增强了对这些技术的理解。

其次，提高了我们阅读文档的能力。我们只有认真阅读并理解微信的开发者指南，才可以正确地调用接口，捕捉用户的行为。

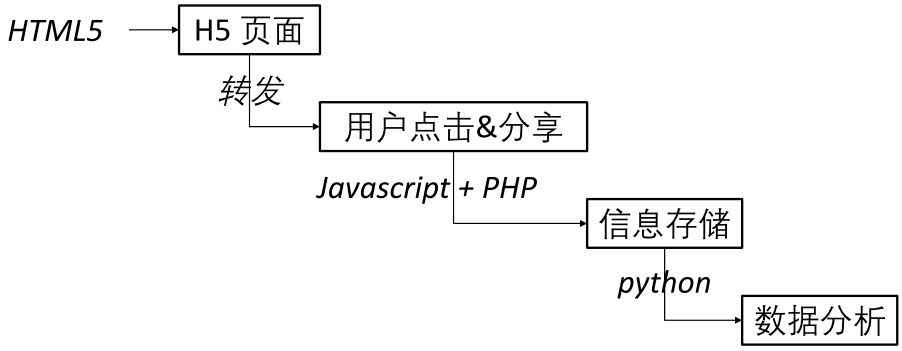
最后，我们获取到了用户与转发信息。通过分析这些信息可以得出各种有价值的数据，比如浏览次数等。

2. 实验原理

2.1. 开发环境

包括 HTML5、JavaScript、PHP 与 python，其中服务器是搭载在阿里云上。

2.2. 系统框图



我们的系统框图如上图所示。用于传播的页面由 H5 编写，在用户点击页面、浏览内容、分享时分别被不同的 JavaScript 接口所捕捉，并将用户 ID、浏览的页面数等信息传送给后端的 PHP 程序，PHP 程序负责将这些数据打上时间戳后传入对应的 SQL 数据库，完成信息的持久化存储。在传播过程结束后，访问数据

库获得原始数据，通过 python 对数据进行分析，得出消息传播路径等信息。

3. 实验内容

3.1. 技术方案

3.1.1. JS-SDK：获取用户信息、监听用户分享事件

微信官方提供了《微信公众平台技术文档》[4]，里面列举了各种可以使用的 JS-SDK 接口。我们主要关心的是其中「微信 JS-SDK 说明文档」部分。它提供了监听用户分享事件和获取用户地理位置的 JavaScript 接口。在调用接口前需要进行相应的配置，如传入公众号 ID、签名等。接口本身的示例如下：

```
wx.onMenuShareTimeline({
    title: '', // 分享标题
    link: '', // 分享链接，该链接域名或路径必须与当前页面对应的公众号 JS 安全域名一致
    imgUrl: '', // 分享图标
    success: function () {
        // 用户确认分享后执行的回调函数
    },
    cancel: function () {
        // 用户取消分享后执行的回调函数
    }
});
```

这个函数会在用户分享到朋友圈的时候被调用，其中的 title 就是分享时候的自定义标题。link 十分关键，它是别人点击这个分享的时候会进入的链接，它的作用就是可以让不同的用户分享后生成的链接不同，使转发路径的记录成为可能。更详细地说，在获取到当前用户 id 后（在 PHP 部分会讲解如何获取），将该 id 加入到 link 中，这样在下一个用户点击这个链接后便可以通过原 id 加新 id 的方式获得一条路径。success 函数在用户分享成功后会被调用，我们可以在这里面记录是哪位用户在什么时间分享了这个文章。

需要特别注意的是，link 后面的注释中说明了「该链接域名或路径必须与当前页面对应的公众号 JS 安全域名一致」，而这句话是新版文档中才有的，组员刚开始查阅的是旧版文档，没有写这句话，于是导致自定义链接总是不成功。调试了好长时间才发现是文档的问题。

除了分享函数之外，JS-SDK 中还有获取用户地理位置信息的函数，但是在尝试获取时页面会弹出一个获取位置请求，在用户点击「允许」后才能得到。用户为了隐私一般不会允许，并且弹窗形式的体验也不尽如人意，因此没有采用。

最后，官方文档后面附了一个 demo[5] 以及源代码 [6]，便于开发者们学习使用 JS-SDK。我们的 JavaScript 代码便是在此基础上编写的。

3.1.2. HTML5：页面展示

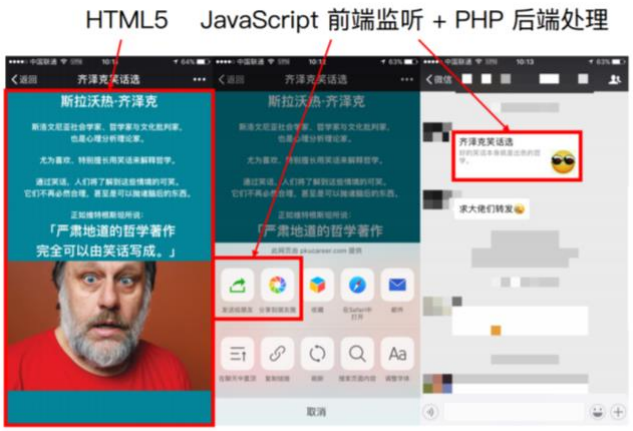
由于本实验对获取浏览页数的需求，我们在网上找到了开源的跨平台滑动控件 iSlider 来帮助我们生成可供翻页浏览的 html 文档。iSlider 在 github 上的网址是 https://github.com/be-fe/iSlide。这次实验中我们所有的 html 和 js 代码都是在一个名为 index.php 的文件里，即包含在了一个 php 程序里。每当用户打开我们的链接时，服务器端运行 index.php，首先通过上述介绍的 JS-SDK 提供的接口获取用户的一些信息，比如个人 ID 以及转发给用户的好友 ID，即一条转发路径。然后 index.php 会生成一个包含 js 代码的 html 文档并发送给用户。用户在翻页欣赏 html 文档内容的同时会触发记录浏览页数的 js 程序，它会通过 XMLHttpRequest() 这一特殊接口将用户浏览的页数发送给位于服务器端的 php 程序 store.php 以进行后续处理。最后当用户欣赏完毕并且要将我们的文档转发给好友时又会触发记录分享次数的 js 程序，它会将用户的个人 ID 以及当前时间等信息发送给 store.php。index.php 中部分 js 代码如下所示：

```
var myslider=new iSlider({
    wrap:'#wrap',
    item:'.item',
    onslide:function (index) {
    var data = new FormData();
    data.append("uid", ""+window.openid);
    data.append("index", ""+index);
    data.append("db", "pages");
    var xhr = new XMLHttpRequest();
    xhr.open('post', 'store.php', true);
    xhr.send(data);
    }
});
console.info(myslider);
```

```
wx.ready(function () {
    wx.onMenuShareAppMessage({
        title: '齐泽克笑话选',
        desc: '好的笑话本身就是出色的哲学。',
        link: 'http://pkucareer.com/GSA/index.php?old='+window.openid,
        imgUrl: '
        https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=14
        1525528989%2C1838838941%26fm%3D214%26gp%3D0.jpg',
        success: function () {
            var data = new FormData();
            data.append("uid", ""+window.openid);
            data.append("db", "share");
            var xhr = new XMLHttpRequest();
            xhr.open('post', 'store.php', true);
            xhr.send(data);
        }
    });
    wx.onMenuShareTimeline({
        title: '齐泽克笑话选',
        link: 'http://pkucareer.com/GSA/index.php?old='+window.openid,
        imgUrl: '
        https://timgsa.baidu.com/timg?image&quality=80&size=b9999_10000&sec=14
        1525528989%2C1838838941%26fm%3D214%26gp%3D0.jpg',
        success: function () {
            var data = new FormData();
            data.append("uid", ""+window.openid);
            data.append("db", "share");
            var xhr = new XMLHttpRequest();
            xhr.open('post', 'store.php', true);
            xhr.send(data);

        }
    });
});
wx.error(function (res) {
    alert(res.errMsg);
});
```

最后关于 html 文档，我们使用的是 iSlider 提供的可供翻页的 html 模板，然后由我们自己组员撰写文章内容，再根据文章内容修改模板使其成为内容丰富、设计精美的微信推送，其目的是为了引发读者阅读的兴趣以增加转发量。显然转发量越多意味着我们的分析结果能更加准确。



hosted by 阿里云

3.1.3. PHP：获取用户 ID、信息存储、数据库数据获取

这次实验中我们主要用到了三个 php 程序，分别是 index.php、store.php 和 fetch.php。它们分别起到了获取用户 ID、信息存储和数据库数据获取的作用。

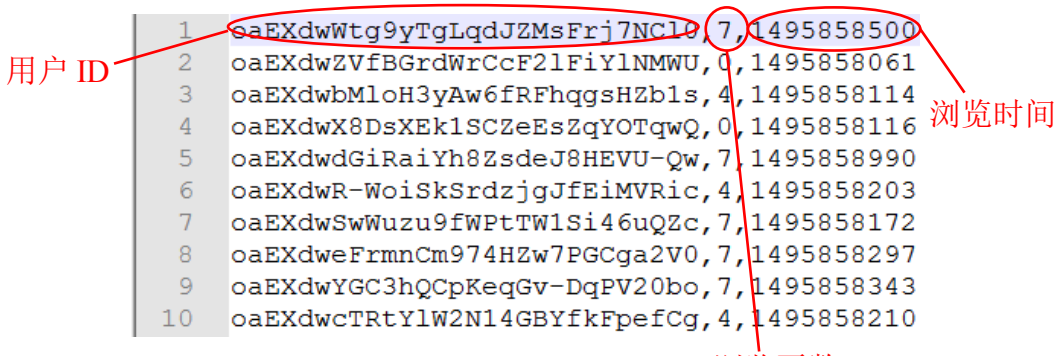
首先来看一下 index.php 中的 php 部分。为了记录转发路径，我们需要区分不同的用户，为每个用户分配一个 ID。这一步骤通过设置 COOKIE 的方式可以完成，不过我们不能确保 COOKIE 一定会留存在浏览器中，会造成用户辨别不准确。此外，微信文档中也提供了获取用户 ID 的方法，因此我们通过微信 api 来获取用户 ID。

拉取用户信息需要两个步骤，首先是通过用户授权获取 code。这一步又分为静默授权和用户手动同意两种。静默授权获取的 code 只能获得用户的 ID，而手动同意获得的 code 可以进一步拉取用户的昵称、所在城市等等详细信息。不过第二种授权在用户打开页面时会有一个显式的授权页面，同样体验很不友好，因此我们决定实现第一种授权方式，用户无感知。第二步便可以通过 code 获取到用户的 openid、也就是唯一的标识了。

在通过 PHP 进行实现时，首先判断 URL 中是否带 code 字段。如果有的话证明已经通过第一步获取到了 code，否则就要重定向到微信指定网址获取 code；然后便是通过 GET 方式向微信指定 uri 发送请求，通过 code 获取到用户的 openid。此 openid 就可以用来标示这个用户。

然后是 store.php。store.php 接收了之前提到过的各种 js 程序发送过来的数据，然后将它们分类并发送到数据库的相应 Table 中。比如将有关用户浏览页数的数据整理并添加到数据库上名为 pages 的 Table 中。值得注意的是，store.php 中对于数据库的操作用到了 php 中的 mysqli 库。

最后是 fetch.php，它的作用是当我们完成用户数据的收集后，我们可以主动运行服务器端的 fetch.php 程序来帮助我们获取位于数据库中的所有数据并将它们以一个统一的格式输出。fetch.php 中对于数据库的操作也用到了 mysqli 库。



获取的部分数据

```php
<?php
if(!empty($_POST['uid'])){
    $dbname = $_POST['db'];

        if($dbname == "share"){
        $db = mysqli_connect("127.0.0.1", "root", "", "db_GSA");
            $query = "INSERT INTO share(User_ID, Time) VALUES('".$_POST['uid']."',".time().")";
            $result = mysqli_query($db, $query);
            if (!$result) {
            print "AError - the query could not be executed";
            $error = mysqli_error($db);
            print "<p>" . $error . "</p>";
            exit;
            }
    )
    else if($dbname == "pages"){
        $db = mysqli_connect("127.0.0.1", "root", "", "db_GSA");
            $query = "SELECT * FROM pages WHERE User_ID='".$_POST['uid']."'";
            $result = mysqli_query($db, $query);
        if (!$result){
            print "EError - the query could not be executed";
            $error = mysqli_error($db);
            print "<p>" . $error . "</p>";
            exit;
        }
            if (mysqli_num_rows($result) == 0) {
            $query = "INSERT INTO pages(User_ID, User_Index, Time) VALUES('".$_POST['uid']."',".$_POST['index'].",".time().")";
                $result = mysqli_query($db, $query);
            if (!$result){
                print "CError - the query could not be executed";
                $error = mysqli_error($db);
                print "<p>" . $error . "</p>";
                exit;
            }
            }
        else{
            $row=mysqli_fetch_array($result);
            $prev_index = $row["User_Index"];
            if($prev_index < $_POST['index']){
                $query = "UPDATE pages SET User_Index=".$_POST['index'].", Time=".time()." WHERE User_ID='".$_POST['uid']."'";
                    $result = mysqli_query($db, $query);
                if (!$result){
                    print "DError - the query could not be executed";
                    $error = mysqli_error($db);
                    print "<p>" . $error . "</p>";
                    exit;
                }
            }
        }
    }
}
?>
```

store.php

```php
<?php
    $db = mysqli_connect("127.0.0.1", "root", "", "db_GSA");
    $query = "SELECT * from pages";
    $result = mysqli_query($db, $query);
    $myfile = fopen("pages.txt", "w") or die("Unable to open file!");
    $num_rows = mysqli_num_rows($result);
    for($row_num=0; $row_num < $num_rows; $row_num ++){
        $row=mysqli_fetch_array($result);
        $txt = $row["User_ID"].",".$row["User_Index"].",".$row["Time"]."\n";
        fwrite($myfile, $txt);
    }
    fclose($myfile);

    $query = "SELECT * from share";
    $result = mysqli_query($db, $query);
    $myfile = fopen("share.txt", "w") or die("Unable to open file!");
    $num_rows = mysqli_num_rows($result);
    for($row_num=0; $row_num < $num_rows; $row_num ++){
        $row=mysqli_fetch_array($result);
        $txt = $row["User_ID"].",".$row["Time"]."\n";
        fwrite($myfile, $txt);
    }
    fclose($myfile);

    $query = "SELECT * from path";
    $result = mysqli_query($db, $query);
    $myfile = fopen("path.txt", "w") or die("Unable to open file!");
    $num_rows = mysqli_num_rows($result);
    for($row_num=0; $row_num < $num_rows; $row_num ++){
        $row=mysqli_fetch_array($result);
        $txt = $row["From_ID"].",".$row["To_ID"].",".$row["Time"]."\n";
        fwrite($myfile, $txt);
    }
    fclose($myfile);
?>
```

fetch.php

3.1.4. python：数据分析

数据分析部分需要对用户转发路径图进行分析。由于 PHP 在数据库中存放的是点对点的转发关系，因此我们以点为基础，将边张成一张图。同时，我们还将转发关系中的时间戳转换为相对时间，以方便估计转发时间等信息。建立完成的图包含了 2259 个节点和 3180 条边。

针对转发路径图，我们借助 networkx 库，从点、边和图的角度分别分析了转发和浏览相关各个因素：

1. 浏览时间
2. 浏览页数
3. 传播时间
4. 传播人数
5. 图的流通性

3.2. 实验步骤

3.2.1. 配置环境

我们使用阿里云来搭建我们的服务器，并绑定为与老师提供的域名。配置 PHP 环境、SQL 服务器。

3.2.2. 实现用户行为监听接口

查阅《微信 JS-SDK 开发者文档》，在页面上添加相应的 JavaScript 接口，并进行测试，确保在用户点击或是分享时可以触发相应的函数调用，可以自定义分享的标题和描述。同时编写 PHP 来获取用户 ID，与 JavaScript 接口结合测试路径的获取。

3.2.3. 编写前后端通信接，实现数据持久化存储

在前端使用 JavaScript 来记录数据，并将采集到的数据通过 GET 或 POST 方法传送给 PHP 脚本文件。PHP 脚本文件分析传来的数据，加上时间戳，传送给 SQL 服务器，实现持久化存储。

3.2.4. 文章撰写

为了获得更大的转发量以提升数据数量与质量以及相应的分析结果，我们需要编写目标人群们所喜闻乐见的推送内容。考虑到本小组成员的微信社交圈以当代大学生群体为主，内容以符合大学生审美与智识需求且轻松愉快、可以缓解学习压力为宜。

从更有效且更全面地统计浏览数据的角度出发，同时也顾及到眼下微信信息阅读的碎片化，我们的推送内容被设计成了分页模式。这就要求推送内容各部分需要相对独立，且每一部分都需要有足够的吸引力，能够让用户保持不断地翻页查看。

综合以上种种考量，小组成员最终决定选取《齐泽克笑话集》中的部分内容作为我们的实验推送。一方面这本书出版不是很久，内容相对新颖，具有吸引力；另一方面，虽然我们最终的推送中因每一页的篇幅限制略去了作者对于每个笑话的所阐发的哲学思考，但相信每一个佐以配图、短小精悍的笑话已经足以引发文化水准相对较高的学生群体们的独立思考，并使其获得一定智力上的愉悦感。最后，内容轻松愉快，也符合我们最初的设想。

最终的统计数据印证了这样的选择是合理的，将近 60%的浏览用户看完了全部的推送内容。在快餐式阅读盛行的今天，这可以说是一个相当不错的成绩，也为各类公众号日后在设计面向学生群体的推送内容时提供了参考。

3.2.5. 点击与转发

点击量与转发量的创造主要依靠两点来实现：小组成员在群聊或者朋友圈中赔本（发红包）赚吆喝来驱动，以及相信我们推送内容的吸引力能够使得用户在浏览之后自发地进行转发。

从最后的统计结果上来看，这两条路径所创造的点击量基本相当。有四个转发吸引了将近两百甚至是两百五十多次的浏览，这很有可能是小组成员在转发之后附赠的红包所带来的效果。除此之外，浏览在一百次及以下的转发数量基本与点击量呈反比，并无太多异常数据出现。可以相信，这部分转发是浏览过后的用户自发产生的行为，并由此形成一条自然的转发链条，因而数据分布符合我们的统计常识。此外，后台的统计表明，最长的转发链条长度为 7，这显然已经远远超出小组成员的控制范围，是推送在用户中自然传播的结果。

3.2.6. 数据分析

首先将 MySQL 中的表本地化，并转成便于处理的 csv 表格，再用 python 脚本脚本加载表中条目，建立转发路径图，分析并得到结果。

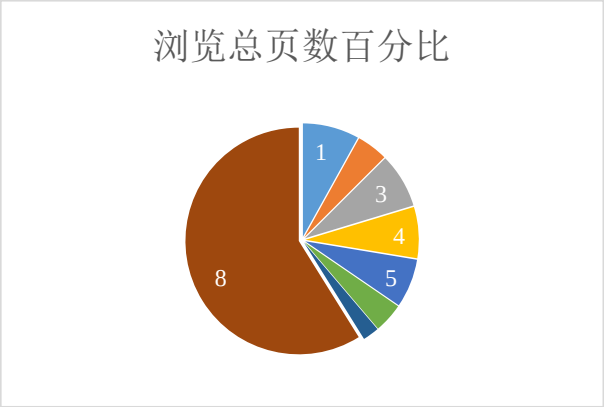
4. 实验结果

4.1. 参数环境

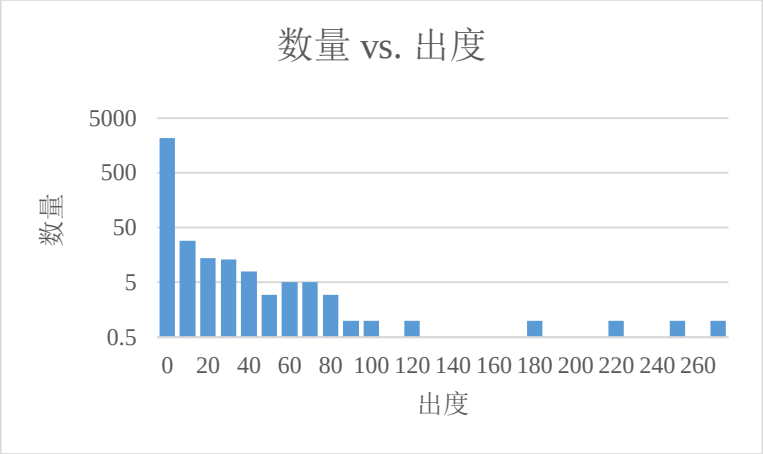
略

4.2. 结果分析

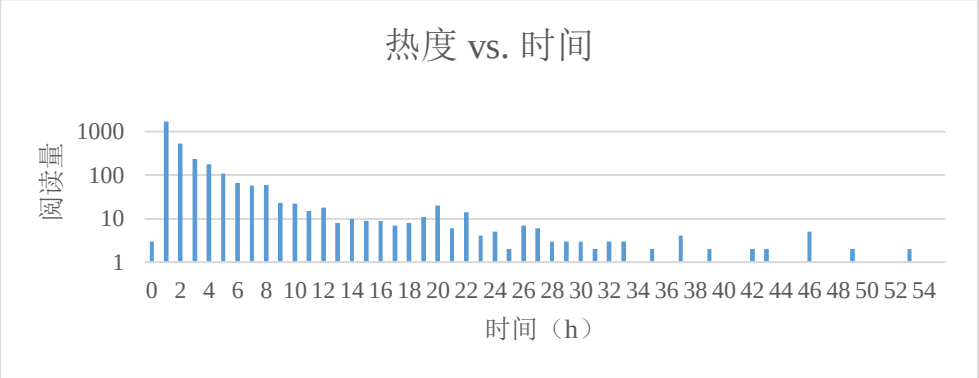
数据分析发现，用户平均浏览页数是 6 页，平均浏览时间是 99s。对于我们总共 8 页的内容而言，这样一个浏览量是相当可观的。平均每页 16s 的浏览时间也和我们的内容符合。对于传播时间而言，平均是 3 个小时左右，最长的则有两天多。这说明用户普遍浏览朋友圈的范围至少包含最近 3 个小时，有部分用户会翻更久远的朋友圈内容。



用户浏览页数分布



用户传播人数分布



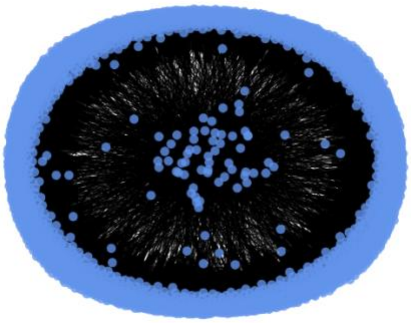
传播时间分布

在传播人数方面，每个人平均将页面转发给了 1.4 个人。这在一定程度上表明，即使是很有趣的内容，单单通过一个人的朋友圈传播，其影响力非常有限。然而也有些人将页面传播给了几十人，最多的有将页面传播给了 266 人。这其中可能有很多是通过群的方式传播的。

通过图的流动性，我们还分析了有效传播的比例。在整张图中，不在任何环中的边有 94%，说明用户在转发的过程中有效传播的比例是相当高的，并不容易出现循环的状态。

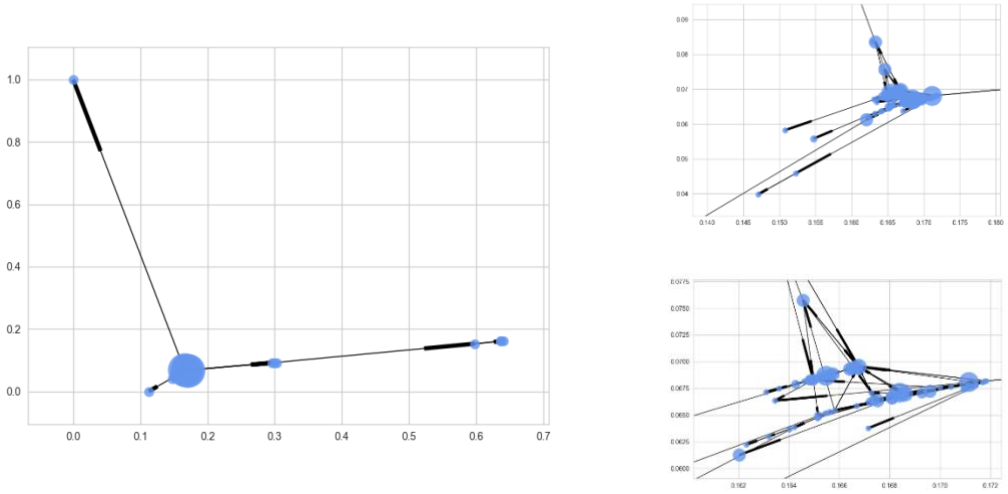
我们对上述因素还做了二变量相关性分析，发现传播人数、传播时间和被传播者浏览页数之间两两线性不相关。

此外，我们基于 matplotlib 库，可视化了整个转发路径图。其中路径长短表示传播时间。大量外层节点都是没有传播的终止节点。



转发路径图

为了更直观的理解整个转发关系，我们去掉了最外层没有传播的节点，对图的核心部分单独做了一张图。我们用点的大小来表示节点在转发过程中距离根节点的跳数，节点越大，说明该节点越接近发出消息的根节点。



转发路径 - 核心部分

可以看出，图中有一块强连通的子图，该图中点与点之间的传播时间都非常短。并且，该图包含四个接近根节点的点，应为本小组的成员。

5. 工作总结

我们在内容的呈现上实现了一个分页展示并记录用户各类行为及浏览数据的框架，以此为基础来制作新的推送并记录与统计相应的浏览与转发数据，十分方便。

但相比于现有的一些付费平台，我们在数据的分析与呈现方面缺少灵活性与系统性，这也是一个可以改进的方向——可以考虑实现一个框架，在后端对于每一个推送生成相应的对象，并在网页端以可视化的方式展示，满足内容发布者的各类查询需求。

附录

[1] 兔展.
http://www.rabbitpre.com/

[2] 人人秀.
https://www.rrxiu.net/

[3] 易企秀.
http://www.eqxiu.com/

[4] 微信公众平台技术文档.
https://mp.weixin.qq.com/wiki

[5] JS-SDK Demo 页面.
http://demo.open.weixin.qq.com/jssdk

[6] JS-SDK Demo 源代码.
http://demo.open.weixin.qq.com/jssdk/sample.zip