

Douban Movie Advanced Search Engine

SHIBI HE, 何石弼 3130000164 求是科学班 13 级 计算机

Abstract

This paper provides a new search engine for common users to search movies on <http://movie.douban.com>. This new search engine is Douban Movie Advanced Search Engine(DMASE). Compared with the default search engine in <http://movie.douban.com>, DMASE is more intelligent, efficient, and powerful. The paper explains how DMASE works and what algorithm DMASE used. To illustrate the advantages of DMASE, this paper lists several examples.

Keywords

Common user, Intelligence, precision

1. Introduction to DMASE

DMASE will get your query and give you a extremely precise and useful results in the blink of an eye. In accordance with your queries, DMASE will show you the movies you probably want to take a view. DMASE is very intelligent, remember, you are forbidden to tell DMASE which direction or category you want to search in. DMASE will find out whether you are searching a movie title, a director, an actor, a description or a genre. You only need to type a few words, and please, let DMASE take care of the other things. DMASE is an advanced search engine, except for a basic searching utility, DMASE also has some advanced functions which will be figured out soon.

DMASE is not designed for analysis, statistics or math purpose. DMASE is not designed for engineers, scientists, administrators, but popular users. Popular users usually search movie by typing a few words, so DMASE is concentrating on search useful, relevant data by phrases or a few words. DMASE is not designed for receiving a long article and using that article to find answers, because common users only want to type less words and still get their results fast and precisely. However, not designed for a long input doesn't mean DMASE could not search by a long article. DMASE is very good at calculating similarity between two papers. For example, if you give DMASE a long article writing about a particular movie scenario, DMASE will quickly find out which movie you are talking about.

DMASE aims at Chinese and English search. Inside DMASE body, there is a mighty parser, it is powerful to do word extraction and sentence segmentation. It could extract phrases, name, date, location from Chinese and English sentences. As you know, most movies in Douban are written in Chinese, and their titles usually consists of both Chinese and English like this: 复仇者联盟 2 : 奥创纪元 Avengers: Age of Ultron, 盗梦空间 Inception. DMASE are able to analyze them perfectly. But inevitably, some other movies are freaks, their information often like this: 名门绅士之缘定芳林 ภาพยนตร์จากเทพ คุณชายรัชชานนท์ (2013) and 跨越彩虹 오버 더 레인보우 (2006) DMASE parser works perfectly when dealing with Chinese and English. And although DMASE is designed for Chinese and English users, in the under layer of the program, it is achieved by unicode character, so basically it still works flawlessly with your Korean, Russian, Japanese, Indian input!

2. The improvement in DMASE

Suppose you, a common user, are going to find some fictions which you may be interested in. So you will search “科幻” on Douban, you will get these results:

你是不是想找 杭州的"科幻"影院?



无锡5D科幻电影院

梅村泰伯二期西大门华联超市向北100米 13912392684 13605231395

搜索更多 "科幻"相关影院



残废科幻 / 歹戔废禾斗么 / Deformity Sci-fi

中国大陆 / 薛鉴羌 / 122分钟 / 科幻 / 犯罪 / 纪录片 / 薛鉴羌 Jianqiang Xue / 汉语普通话

★★★★★ 7.3 (105人评价)



科幻电影与未来时代

中国大陆 / 张东林 / 20分钟 / 纪录片 / 汉语普通话

(少于10人评价)



科幻大师

2007-08-04(美国) / 史蒂芬·霍金 / 朱迪·戴维森 / 约翰·赫特 / 布莱恩·丹内利 / 詹姆斯·丹顿 / 伊丽莎白·霍尔姆 / 金伯莉·伊丽丝 / 安·海切 Anne Heche / 马尔科姆·麦克道威尔 / 小克利夫顿·克林斯 / 西恩·奥斯汀 / 詹姆斯·克伦威尔 / ...

★★★★★ 7.8 (1116人评价)



科学频道 科幻科学：不可能的物理学 / 科学频道 科幻科学：不可能的物理学

The top answer is “你是不是要找杭州的“科幻影院””. Let's put aside how stupid this top answer is, just follow its suggestion by clicking. Then you will get:

杭州-电影院 [切换城市]

科幻 搜影院

区域: 全部 江干区 萧山区 上城区 拱墅区 余杭区 下城区 西湖区 临安 富阳 滨江区 桐庐县 建德 淳安县

定位到附近的影院

没有你要找的, 你可以换个关键词再试试。

> 去全部

杭州观影



Oh no. It's a dead end!

Now I assume you ignored the ridiculous top answer, but look at here:

你是不是想找 杭州的“科幻”影院？



无锡5D科幻电影院 影院

梅村泰伯二期西大门华联超市向北100米 13912392684 13605231395

搜索更多“科幻”相关影院



残废科幻 / 歹戔废禾斗么ㄣ / Deformity Sci-fi

中国大陆 / 薛鉴羌 / 122分钟 / 科幻 / 犯罪 / 纪录片 / 薛鉴羌 Jianqiang Xue / 汉语普通话

★★★★★ 7.3 (105人评价)



科幻电影与未来时代

中国大陆 / 张东林 / 20分钟 / 纪录片 / 汉语普通话

(少于10人评价)



科幻大师

2007-08-04(美国) / 史蒂芬·霍金 / 朱迪·戴维斯 / 约翰·赫特 / 布莱恩·丹内利 / 詹姆斯·丹顿 / 伊丽莎白·霍尔姆 / 金伯莉·伊丽丝 / 安·海切 Anne Heche / 马尔科姆·麦克道威尔 / 小克利夫顿·克林斯 / 西恩·奥斯汀 / 詹姆斯·克伦威尔 / ...

★★★★★ 7.6 (1116人评价)



科学频道 科幻科学：不可能的物理学 / 科学频道 科幻科学：不可能的物理学

I am one hundred percent sure that you were not looking for this movie. Just look at this movie, it got no comment, no photo and no rating. How is that possible a popular user wants to watch that kind of movie?

We are users. We are neither a movie librarian nor a Douban movie manager. We are users! So let those kind of movies stay in the dust bin and never bother us again, because we, the popular users, do not need them.

Apparently the search engine on Douban is based on matching the movie title with the given input, then showing users the matched results.

Attention here, the “matched movies” means partially matched or let me put it another way: the title of the movie contains the input word (“科幻电影与未来时代” contains “科幻”). Further more, if you are searching the word: “科幻电影与未来时代” instead of “科幻”, in this case, no matter how unpopular, unfamiliar the movie “科幻电影与未来时代” is, “科幻电影与未来时代” should be the top answer because the user typed exactly the movie’s title. When user typed words like “科幻”, in tens of thousands of movies, the partially title matched results may not be the good answers.

搜索更多“科幻”相关影院

藤子・F・不二雄的科幻短篇剧场 / Fujiko F Fujio no SF Tanpen Theater

1990-04-01 / 草尾毅 / 佐々木望 / 古谷徹 / 岩田光央 / 飛田展男 / 古川登志夫 / 龍田直樹 / 玉川砂己子 / 山口京子 / 藤波圭一 / 高山みなみ / 日本 / 望月智充 / 藤子・F・不二雄的科幻短篇剧场 / 日语 / 草尾毅 / 佐々木望 / 古谷徹 / 岩田光央...

(少于10人评价)

科幻故事

1996-04-12 / David Duchovny / Gillian Anderson / 美国 / Rob Bowman / Argentina: 60 分钟 / 90 分钟 / 剧情 / 悬疑 / 科幻 / 惊悚 / 英语

(14人评价)

科幻世纪 / A Century of Sci-Fi

雷·布萊德伯里 / 尤·伯連納 / 凱文·科斯特納 / 美国 / Ted Newsom / 科幻世纪 / 纪录片 / 英语

(少于10人评价)

科幻玩过瘾 / 科幻玩过瘾

2006-11-03 / David Mitchell / D.C. Douglas / Robert Patrick / Kevin Warwick / 英国 / M.J. Longstrech / Turner / 科幻玩过瘾 / 英语

(少于10人评价)

科幻剧场 / 科幻剧场

1955-1957 / Truman Bradley / 美国 / Herbert L. Strock / Jack Arnold / Leon Benson / 25 分钟 (78 episodes) / 科幻剧场 / 剧情 / 科幻 / Eric Freiwald / Laurence Heath / 英语

(目前无人评价)

登月真相的背后：比科幻小说更高奇 / The Truth Behind the Moon Landings

巴兹·奥德林 / Marcus Allen / Bill Kaysing / Neil Morrissey / 英国 / 加拿大 / Canada / Virginia Quinn / 50分钟 / 登月真相的背后：比科幻小说更高奇 / 纪录片

> 添加影人 科幻

相关搜索

> 搜索科幻的图书

> 搜索科幻的音符

> 搜索科幻的舞蹈

OpenSearch: RSS



page2

Let us still focus on the keyword:“科幻”.



时空恋旅人 / 时空旅恋人 / 回到最爱的一天(港) [可播放]

2013-06-27(爱丁堡电影节) / 2013-09-04(英国) / 多姆纳尔·格利森 / 瑞秋·麦克亚当斯 / 比尔·奈伊 / 莉迪亚·威尔逊 / 汤姆·霍兰德 / 琳赛·邓肯 / 玛格特·罗比 / 凡妮莎·柯比 / 李·阿斯奎斯-柯 / 凯瑟琳·斯戴曼 / 英国 / www.abouttimemovie.com...

★★★★★ 8.5 (109478人评价)



前目的地 / 宿命论(港) / 逆时空狙击(港) [可播放]

2014-03-08(西南偏南电影节) / 2015-01-09(中国大陆/美国) / 伊桑·霍克 / 莎拉·斯努克 / 诺亚·泰勒 / 弗雷娅·斯塔福 / 伊莉斯·詹森 / 凯特·沃尔夫 / 迈德琳·怀斯特 / 亚历克斯·史密斯 / 克里斯托弗·卡比 / 罗伯·詹金斯 / 艾丽西娅·帕夫利斯...

★★★★★ 7.6 (53316人评价)

<前页 1 2 3 4 5 6 7 8 9 ... 431 432 后页> (共6466条)

When we turn to page 3, we will find these results:



星际穿越 / 星际启示录(港) / 星际效应(台) [可播放]

2014-11-07(美国) / 2014-11-12(中国大陆) / 马修·麦康纳 / 安妮·海瑟薇 / 杰西卡·查斯坦 / 迈克尔·凯恩 / 麦肯吉·弗依 / 蒂莫西·柴勒梅德 / 约翰·利特高 / 韦斯·本特利 / 大卫·吉雅西 / 比尔·欧文 / 马特·达蒙 / 卡西·阿弗莱克 / 托弗·戈瑞斯...

★★★★★ 9.1 (313768人评价)



复仇者联盟2：奥创纪元 / 复仇者联盟2 / 复仇者联盟：奥创时代

2015-05-01(美国) / 2015-05-12(中国大陆) / 小罗伯特·唐尼 / 克里斯·海姆斯沃斯 / 马克·鲁弗洛 / 克里斯·埃文斯 / 斯嘉丽·约翰逊 / 杰瑞米·雷纳 / 詹姆斯·斯派德 / 塞缪尔·杰克逊 / 唐·钱德尔 / 亚伦·泰勒-约翰逊 / 伊丽莎白·奥尔森 / ...

★★★★★ 7.1 (124094人评价)



分歧者2：绝地反击 / 叛乱者：强权终结(港) / 分歧者2：叛乱者(台)

2015-03-20(美国) / 2015-06-19(中国大陆) / 谢琳·伍德蕾 / 提奥·詹姆斯 / 凯特·温丝莱特 / 奥克塔维亚·斯宾瑟 / 杰·科特尼 / 佐伊·克罗维兹 / 迈尔斯·特勒 / 安塞尔·艾尔高特 / 李美琪 / 娜奥米·沃茨 / 梅奇·费法 / 贾斯汀·利克 / 本·劳埃德-休斯...

★★★★★ 5.8 (15417人评价)



彗星来的那一夜 / 相干性 / 相干效应 [可播放]

2013-09-19(奥斯汀奇幻电影节) / 2014-08-06(美国) / 艾米丽·芭尔多尼 / 莫瑞·史特林 / 尼古拉斯·布兰登 / 伊丽莎白·格瑞斯 / 亚历克斯·马努吉安 / 劳伦·马赫 / 雨果·阿姆斯特朗 / 劳伦·斯卡法莉娅 / 美国 / 英国 / 詹姆斯·沃德·布柯特...

★★★★★ 8.3 (73737人评价)



超感八人组 第一季 / 超感猎杀 / 超感八人

2015-06-05(美国) / 米格尔·安赫尔·西尔维斯特 / 杰米·克莱顿 / 布莱恩·J·史密斯 / 裴斗娜 / 阿梅尔·艾米恩 / 塔彭丝·米德尔顿 / 马克思·雷迈特 / 蒂娜·德赛 / 弗莉玛·阿吉曼 / 纳威恩·安德利维斯 / 达丽尔·汉纳 / 豪威·约翰逊 / 亚当·沙皮罗...

★★★★★ 8.7 (11642人评价)

These movies are exactly the results that common users wanted to view. When a big and unspecific word like "科幻" typed, users are expected to find some good and popular movies. Usually people will type a more detailed sentence to find a particular movie which may not be very popular.

Now the problem is how to search both precisely and intelligently.

Improving the engine's intelligence is exactly what DMASE is trying to achieve. One way is the recommendation system: after logged in my account, the search engine will give me the answer to "科幻" according to my user habit. But DMASE won't choose that, because DMASE is a general or universal search engine, designed for popular and common users, DMASE has to give users good answers no matter whether users have a movie account, watching history or not.

People care a lot about the difference between recommendation system and search system. Recommendation system has its own defects. It always get lost, misled, wrong guided, so pushing completely irrelevant suggestion happens. DMASE is not going to be a recommendation system.

3. Some demos of DMASE

If we search "科幻", the results in left side are computed from DMASE.


```
{
  "a_title": "纵横四海 縱橫四海",
  "b_alias": "C",
  "url": "http://movie.douban.com/subject/1297570/"
},
{
  "a_title": "变相怪杰 The Mask",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/1292326/"
},
{
  "a_title": "暴力街区 Banlieue 13",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/1306982/"
},
{
  "a_title": "吸血鬼猎人D Vampire Hunter D: Bloodlust",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/4914656/"
},
{
  "a_title": "猫在巴黎 Une vie de chat",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/1921465/"
},
{
  "a_title": "火炬木小组 第一季 Torchwood Season 1",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/3630615/"
},
{
  "a_title": "生命的形状 PBS: Shape of Life",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/1299502/"
},
{
  "a_title": "霹雳火 霹靂火",
  "b_alias": "Thunderbolt",
  "url": "http://movie.douban.com/subject/4100696/"
},
{
  "a_title": "美丽密令",
  "b_alias": "Beauty on the Mind",
  "url": "http://movie.douban.com/subject/1300714/"
},
{
  "a_title": "飞龙猛将 飛龍猛將",
  "b_alias": "Dragon Storm",
  "url": "http://movie.douban.com/subject/2338069/"
},
{
  "a_title": "亲密如贼 Thick as Thieves",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/1307403/"
},
{
  "a_title": "飞狗巴迪5: 排球健将 Air Bud: Spikes vs. Meanie",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/2129389/"
},
{
  "a_title": "假面骑士KABUTO 仮面ライダーカブト",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/1295185/"
},
{
  "a_title": "乌龟和兔子 The Tortoise and the Hare",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/2269032/"
},
{
  "a_title": "地下拳击场 Fighting",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/1294467/"
},
{
  "a_title": "皇家师姐",
  "b_alias": "Yes, Madam",
  "url": "http://movie.douban.com/subject/6738637/"
},
{
  "a_title": "火线干探之战火的洗礼 Alarm für Cobra 11 - Die Autobahnpolizei",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/22991965/"
},
{
  "a_title": "金麦侦探社 第一季 King and Maxwell",
  "b_alias": "T",
  "url": "http://movie.douban.com/subject/3215511/"
}
```



If we search “冒险 英雄”, the results in left side are computed from DMASE.

{
"a_title": "无敌破坏王 Wreck-It Ralph", "b_ali
http://movie.douban.com/subject/11026735/ }
{"a_title": "超能陆战队 Big Hero 6", "b_ali
http://movie.douban.com/subject/1295398/ }
{"a_title": "七武士 七人の侍", "b_aliases": "七
http://movie.douban.com/subject/1478186/ }
{"a_title": "射雕英雄传 射雕英雄传", "b_ali
http://movie.douban.com/subject/2223586/ }
{"a_title": "特种部队：眼镜蛇的崛起 G.I. Joe
http://movie.douban.com/subject/3927734/ }
{"a_title": "乐高大师电影 The Lego Movie", "b_
http://movie.douban.com/subject/2049435/ }
{"a_title": "超人：钢铁之躯 Man of Steel", "
http://movie.douban.com/subject/3263814/ }
{"a_title": "森林战士 Epic", "b_aliases": "绿国
http://movie.douban.com/subject/1294671/ }
{"a_title": "中华英雄 中華英雄", "b_aliases": "
http://movie.douban.com/subject/21349175/ }
{"a_title": "乐高蝙蝠侠大电影：DC英雄集结 LE
http://movie.douban.com/subject/2090440/ }
{"a_title": "月球大冒险 Fly Me to the Moon", "
http://movie.douban.com/subject/1866479/ }
{"a_title": "复仇者联盟 The Avengers", "b_al
http://movie.douban.com/subject/7057975/ }
{"a_title": "考拉大冒险 코알라 키디: 영웅의
http://movie.douban.com/subject/1281586/ }
{"a_title": "海底总动员 Finding Memo", "b_al
http://movie.douban.com/subject/1293764/ }
{"a_title": "与狼共舞 Dances with Wolves", "
http://movie.douban.com/subject/1437342/ }
{"a_title": "冰川时代2：融冰之战 Ice Age: Th
http://movie.douban.com/subject/1292925/ }
{"a_title": "伴我同行 Stand by Me", "b_ali
http://movie.douban.com/subject/2133323/ }
{"a_title": "白日梦想家 The Secret Life of W
http://movie.douban.com/subject/2191581/ }
{"a_title": "怪物史瑞克 Shrek", "b_aliases": "

考拉大冒险 / 考拉小子：英雄的诞生 / 考拉：英雄的诞生
2012-01-12(美国) / 2014-05-01(中国大陆) / 导演: 比尔·柏拉夫 / 艾伦·卡明 / 蒂姆·克雷斯 / 克里斯·埃利斯 / 德里克·李 / 查理·罗伯茨 / 布赖恩·史密斯 / 詹姆斯·沃克 / 菲尔·霍洛威特 / 珍妮·斯皮尔 / 内尔·德·格罗斯

★★★★☆ 5.4 (698人评价)

守望先锋 / 极地恶灵 Ji di huang ling
主创: 唐纳德·桑迪 / 斯科特·考文 / 斯科特·考文 / 张晋 / 张晋 / 张晋 / 90分钟 / 守望先锋 / 动作 / 冒险 / 科幻
何浩 Yu-Hao Wang / 粤语 / 普通话

★★★★☆ 4.3 (309人评价)

救难小英雄 / 救难小英雄 / 神勇小英雄
1977-12-22 (美国) / Art Stevens / John Lounsbey / (伍尔夫雷德) (Wolfgang Retherman) / 78分
钟 / 救难小英雄 / 动画 / 冒险 / 犯罪 / 喜剧 / 家庭 / Burny Mattinson / David Michener / Dick Seabel / 英语

★★★★★ 7.8 (193人评价)

救难小英雄-澳洲历险记 / 救难小英雄-澳洲历险记
1990-11-16 (美国) / Hensel Butty / 迈克·加布里埃尔 (Mike Gabrier) / USA: 77分 / 救难小英雄-澳洲历险记 / 动画 / 冒险 / 犯罪 / 家庭 / 奇幻 / 悬疑 / Byron Simpson / Jim Cox (山姆·肖) / 乱入
Joe Ramo / 英语

★★★★★ 7.3 (165人评价)

麦斯卡：寻找英雄 / 麦斯卡少女勇闯龙洞 / 凯瑟琳寻找英雄
2015-1-22(美国) / 2015-3-27(德国) / Melanie Stone / Adam Johnson / Jake Stormoen / Nicola Posner / Christopher Rodin Miller / 凯莉·安托 / 安妮 / Karen Carpenter / 93分钟 / 麦斯卡寻找英雄 / 动作 / 奇幻 / 冒险 / Anne K. Black / Jason Faller / Kyran Griffin / 英语

★★★☆☆ 3.9 (37人评价)

邪恶湾城2尸代英雄 / 邪恶湾城尸代英雄 / 英雄时代的恶棍
希腊 / Yorgos Noussias / 88分钟 / 邪恶湾城2尸代英雄 / 冒险 / 喜剧 / 恐怖 / 希腊语

(22人评价)

少数民族英雄们 / 少数民族英雄们
2005-11-05 / Dana Snyder / Keith Law / 尼克·普里查德 / 美国 / USA / Adam De La Peña / Peter Girard / 少数民族英雄们 / 英语 / English

(少于10人评价)

超级英雄军团 第一季 / 超级英雄军团 第一季
Yuri Lowenthal / Andy Milder / Karl Whelan / 美国 / Brandon Vietti / Ben Jones / Tim Matney / Lauren Montgomery / James Tucker / Scott Jerads / 22分钟 / 超级英雄军团 / 动作 / 科幻 / 动画 / 英语 / Yuri Lowenthal / Andy Milder / Karl Whelan

(评价人数不足)

怒河英雄队 / 怒河英雄队 / 白日的咆哮：大地的哭声
美国 / Margo Blue / Catherine Cyran / 怒河英雄队 / 动作 / 冒险 / 家庭 / Margo Blue / Catherine Cyran / 英语

(少于10人评价)

**"a_title": "童话镇 第二季 Once Upon a Time Season 2",
http://movie.douban.com/subject/2372452/**

**"a_title": "咕咚咕咚魔法阵 魔法陣グルグル", "b_alias": "魔法阵咕噜咕噜",
http://movie.douban.com/subject/4816603/**

**"a_title": "魔境仙踪 Oz: The Great and Powerful",
http://movie.douban.com/subject/3251306/**

**"a_title": "魔法禁书目录 とある魔術の禁書目録",
http://movie.douban.com/subject/2150453/**

**"a_title": "境界奇谭 第一季 Tales From The Crypt",
http://movie.douban.com/subject/1960296/**

**"a_title": "宝葫芦的秘密", "b_alias": "飞天小葫芦",
http://movie.douban.com/subject/1296154/**

**"a_title": "墨林 Merlin", "b_alias": "终极/最终",
http://movie.douban.com/subject/2223153/**

**"a_title": "大魔法师", "b_alias": "Funie", "c_year": 2009,
http://movie.douban.com/subject/2132425/**

**"a_title": "探索者传说 第一季 Legend of the Seeker",
http://movie.douban.com/subject/4185683/**

**"a_title": "美丽生灵 Beautiful Creatures", "b_alias": "魔法仙境",
http://movie.douban.com/subject/4139823/**

**"a_title": "探索者传说 第二季 Legend of the Seeker: Book Two",
http://movie.douban.com/subject/1401531/**

**"a_title": "地海传说 Legend of Earthsea", "b_alias": "地海英雄传",
http://movie.douban.com/subject/24694723/**

**"a_title": "奇境传说 Once Upon a Time in Wonderland",
http://movie.douban.com/subject/4451480/**

**"a_title": "情爱魔力 마법의 성", "b_alias": "魔法爱情",
http://movie.douban.com/subject/3543704/**

**"a_title": "不老传说", "b_alias": "", "c_year": 2009,
http://movie.douban.com/subject/4006470/**

**"a_title": "妖精的尾巴 フェアリーテイル", "b_alias": "魔导少年",
http://movie.douban.com/subject/1482058/**

**"a_title": "小狗多戈尔 Boogal", "b_alias": "",
http://movie.douban.com/subject/25727263/**


**"a_title": "极黑的布伦希尔特 极黒のブリュンヒルデ",
http://movie.douban.com/subject/3475347/**


豆瓣电影


魔法传说


新闻 & 热搜 电视剧 排行榜 分类 影评 预告片 问答


你是不喜欢魔法？

 **咕咚咕咚魔法阵·心魔传说 / 咕咚咕咚魔法阵·心魔传说**
2009-04-04—2009-12-25 / 流木富士子 / 陆合贵一 / 日本 / 高木孝 / 22分钟 / 咕咚咕咚魔法阵·心魔传说 / 动画 / 三井洋行 / 日语
★★★★★ 7.3 (187人评价)

 **魔法禁书目录 / 某魔法的禁书目录 / 传说中中国的禁书目录**
2008-10-04 (日本) / 前田敦 / 井口祐介 / 佐藤利奈 / 杉并真央 / 冈本信彦 / 阪田麻衣子 / 伊藤静 / 日永 / 伊藤静 / 24分钟 / 魔法禁书目录 / 动画 / 富田 / 藤原 / 曾田 / 饭沼 / 科幻 / 水上清良 / 高桥 Minami / 日语 / 日语版 / 井口祐介
★★★★★ 7.6 (807人评价)

 **地海传说 / 地海传说**
2004-12-13 / Shivan Anandah 桑德拉艾西露 / Kristin Knuck 凯拉托瓦黛尔 / Isabella Rossellini 伊莎贝拉·罗塞里尼 / Danny Glover 丹尼·格洛弗 / 英语 / Robert Lieberman / 地海传说 / 剧情 / 奇幻 / 冒险 / Ursula K. Le Guin / Gavin Scott / 英语
★★★★★ 8.3 (108人评价)

 **最终 / 终极 / 终极魔法传说**
1999-11-13 / 山崎 浩史 / 米歇尔·道奇斯 Miranda Richardson / 詹姆斯·肯特 James Bonham Carter / 伊万娜·巴克里 / 马丁·斯科帕里 / Paul Curran / 英语 / 葡萄牙语 / 意大利语 / 西班牙语 / 法语 / 德语 / 波兰 / 爱情 / 奇幻 / 冒险 / David Deviers / Edward Roman...
★★★★★ 7.6 (785人评价)

 **魔界奇谭 第一季 / 地穴传说 / 魔法魔法传说**
1989-05-10(美国) / 约翰·卡塞尔 / 英国 / HBO documentary / 魔界奇谭 / 25分钟 / 魔界奇谭 / 犯罪 / 恐怖 / 悬疑 / Scott Nimeroff / William M. Gaines / 英语
★★★★★ 8.4 (520人评价)

© 2005 - 2015 douban.com, all rights reserved 关于豆瓣 | 在豆瓣工作

Both the precision and the intelligence of DMASE are apparently far better than the default search engine in <http://movie.douban.com>.

For more details, please connect to ZJUWlan, our searching web sever is here:

<http://10.214.0.195:10000/>



4. Search engine design

4.1 Keywords generation

First I will show some basic algorithm in searching engine:

1. Term Frequency Weight

The log frequency weight of term t in d is defined as follows

$$w_{t,d} = \begin{cases} 1 + \log_{10} \text{tf}_{t,d} & \text{if } \text{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

2. Idf Weight

The document frequency df_t is defined as the number of documents that t occurs in W . We define the idf weight of term t as follows:

$$\text{idf}_t = \log_{10} \frac{N}{\text{df}_t}$$

3. Tf-idf Weight

The tf-idf weight of a term is the product of its tf weight and its idf weight:

$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t}$$

4. Cosine Similarity between Query and Document

$$\cos(\vec{q}, \vec{d}) = \text{SIM}(\vec{q}, \vec{d}) = \frac{\vec{q}}{|\vec{q}|} \cdot \frac{\vec{d}}{|\vec{d}|} = \sum_{i=1}^{|V|} \frac{q_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2}} \cdot \frac{d_i}{\sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

let us consider the Cosine Similarity between Query and Document. Given that the different parts of the information in a movie have different significance(the movie title is apparently more important than the movie description), this algorithm is not suitable for movie search engine.

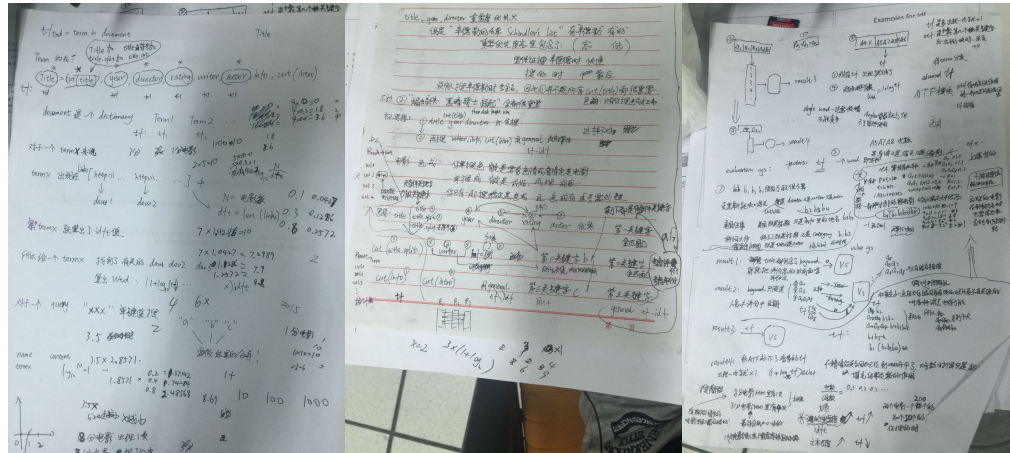
Usually the Cosine Similarity between Query and Document is powerful in general text search.

So the features of a movie search engine are:

1. Common users only want to type less words and still get their results fast and precisely.

2. The information of a movie has different priorities(The keywords in the title and the director is much more significant. But those in movie description are less important).

Based on these features, DMASE is aim at searching extremely precise on a single keyword. Then I built several models for DMASE, let us have a look.



Finally I designed my searching structure using keyword hierarchy.

Generate keywords

Title: Title: a1. Title.split(): a2 reduceSign(title.split()): a2

“食神”“宿主 The Host” “色，戒”

Alias[]: a1. Alias[].split(): a2 reduceSign(alias[].split()): a2

Year[0]: Year[0]: a4.

Director[]: Director[] and split(): a7

***Rating[]:** *Rating[]: a5

Actor[]: Actor and split(): a6

Writer[]: writer and split(): a6

Important roles are first five actor and director

Cut_list(title.split()): b1 Only cut when new words generated

Cut_list(alias[].split()): b1

Cut_list(director[]): b2

Cut_list(actor[]): b2

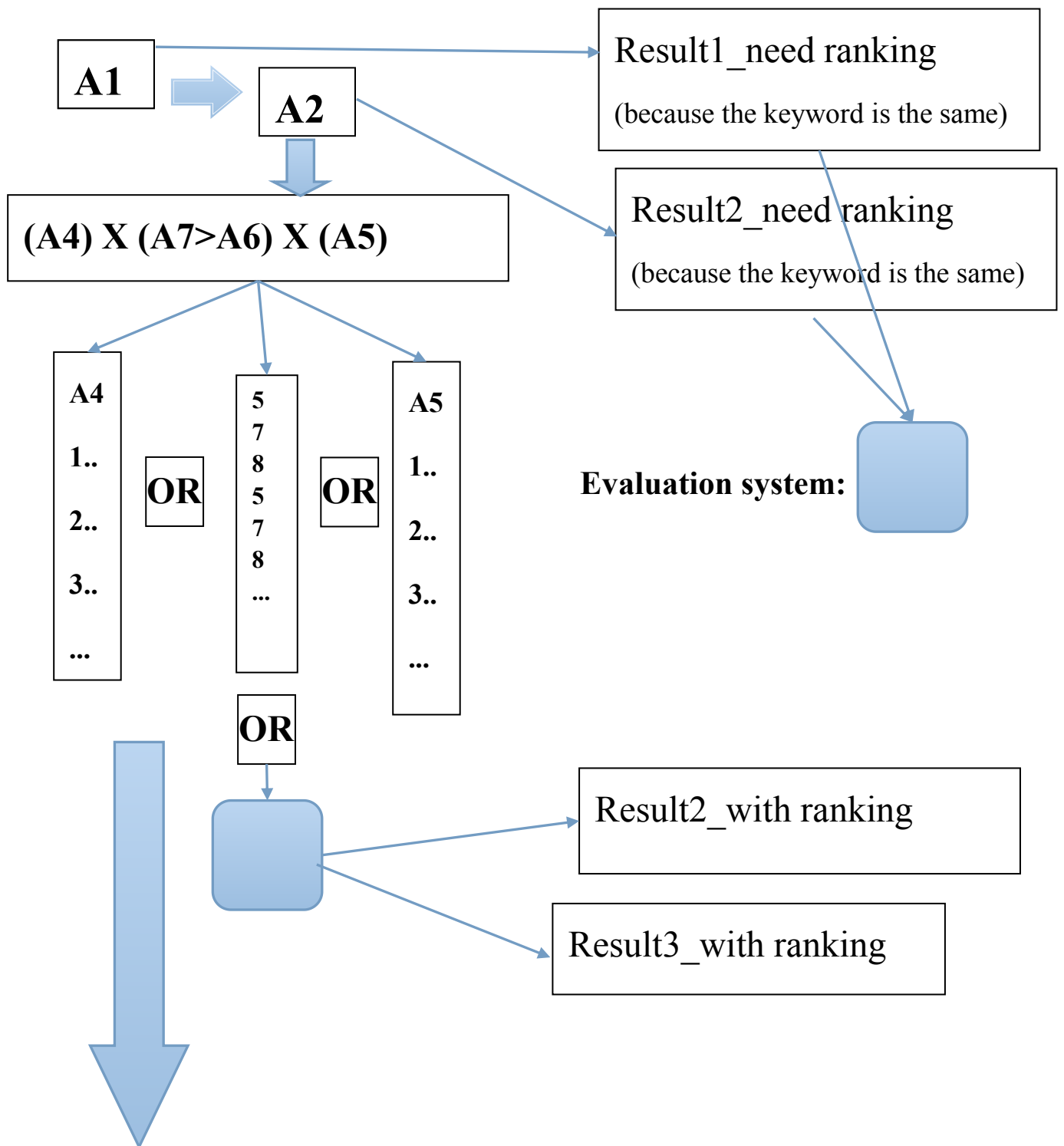
Cut_list(writer[]): b2

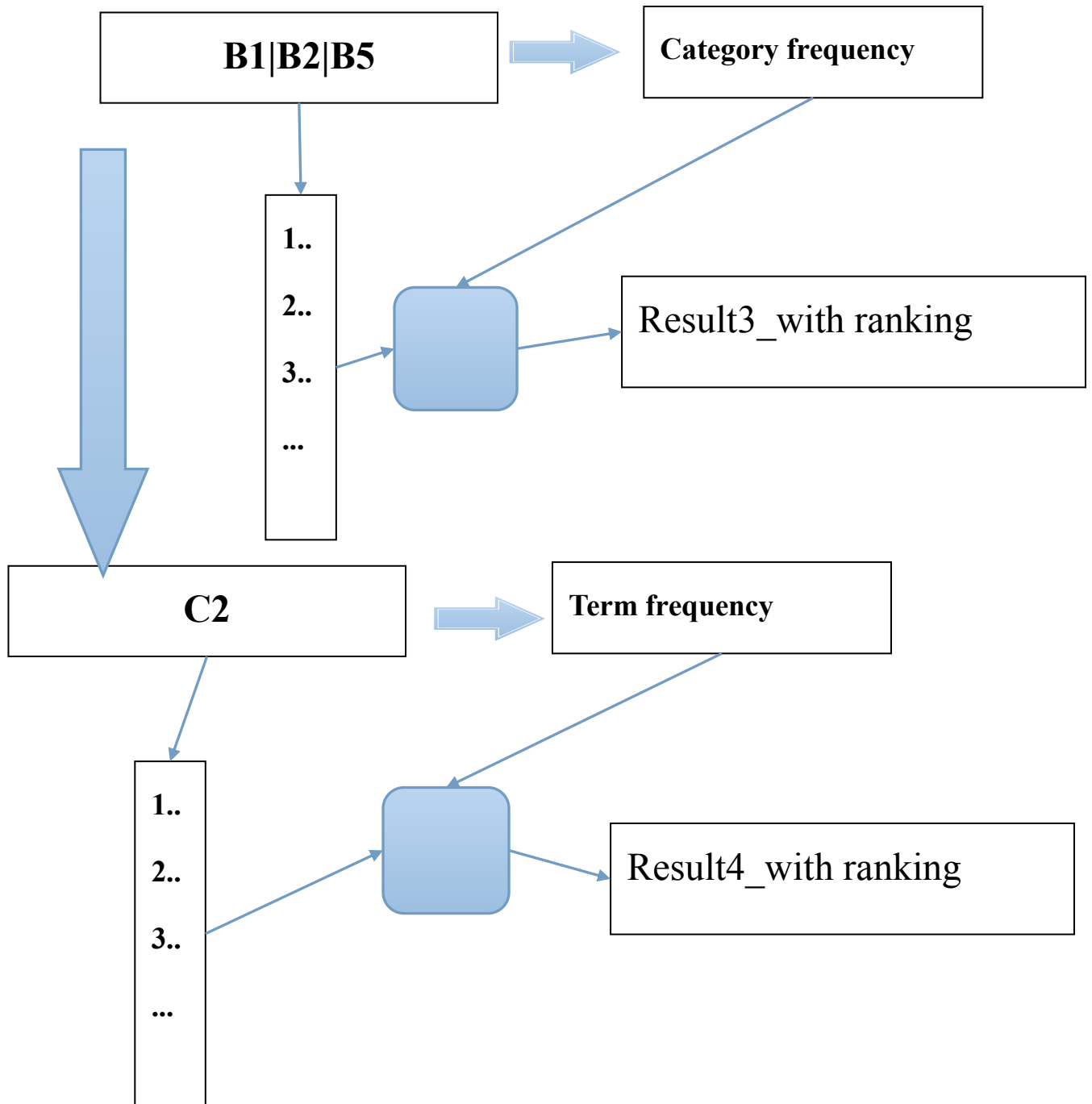
Info[]: b5

Cut_list(info[]): c1 united with b5

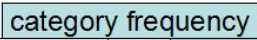
Cut_list(intro[]): c2

The number a1, a2, b1, b2 are the priorities of the keywords. DMASE will search the highest priority keywords first, then the lower priority keywords, finally the least important keywords.





complete matching



term frequency

4.2.1 rating and comments

people who commented on this movie while *the Beethoven Symphony No.9* only has 84 people who has made comments.



肖申克的救赎 The Shawshank Redemption (1994)

★★★★★ 9.6

(623860人评价)

81.0%

★★★★★ 10.0%

★★★★★ 2.2%

★★★★★ 0.1%

★★★★★ 0.1%

导演: 弗兰克·德拉邦特

编剧: 弗兰克·德拉邦特 / 斯蒂芬·金

主演: 蒂姆·罗宾斯 / 摩根·弗里曼 / 鲍勃·冈顿 / 爱德华·诺顿 / 克林特·伊斯特伍德 / 克兰西·布朗 / 更多...

类型: 剧情 / 犯罪

制片国家/地区: 美国

语言: 英语

上映日期: 1994-09-10(多伦多电影节) / 1994-10-14

片长: 142分钟

又名: 月黑风高(港) / 刺激1995(台) / 地牢直直 / 铁窗岁月 / 肖申克的救赎

IMDb链接: [tt0111161](#)

好于 99% 剧情片

好于 99% 犯罪片

top famous movies usually have 10^5 comments;

recent common movies usually have 10^3 comments;

Considering the number of comments is significant, I designed the final rating for DMASE, combined both rating and comment number.

Final rating = rating(0~10)*log(comments_amount) (1~ 60)

4.2.2 category frequency

If a keyword is both in a movie actor's name and in that movie's title, then that movie will probably be the user's first choice. The category frequency is trying to find a keyword that appears in multiple categories.

The category frequency algorithm was implemented in B1 B2 B5, which are cut titles, cut names and movie information. For example, if we search "美国", in the movie "美国往事", "美国" is a part of the cut title, and "美国" is the region in movie information, then the movie "美国往事" will be more significant in the ranking system, because it has strong connection with the word "美国". Another example is "霍金传". "霍金" both appears in title and names.

Cut_list(title.split()): b1

Cut_list(alias[],split()): b1

Cut_list(director[]): b2

Cut_list(actor[]): b2

Cut_list(writer[]): b2

Info[]: b5

Cut_list(info[]): c1 united with b5

B1 B2 B5

In 4.2.1, rating and comments system will always put the famous movies on the top of the results. However, sometimes a keyword is so important that it can identify some close related movies immediately, in this case rating and comments system is not efficient because it only considered one principle: good movie first.

In category frequency algorithm, I am trying to find the movies that are closed related with the keywords, even if those movies are not famous or popular. The simplest way is to use the score in rating and comment system to multiply a coefficient. If the keyword appears in only one category, the coefficient is 1. If the keyword appears in two or three categories, we can make the coefficient into 1.3 or 1.5. But this method is not good enough. An unpopular movie may has few comments, so its score in rating and comment system is very low(20 or 30 of 60). Whatever the coefficient is, the movie will always in the bottom of the ranking results compared with the famous and popular movies which always get 50 or even 55.

My solution is $((\frac{60}{x} - 1) * coefficient + 1) * x$ where x is the score from rating and comment system

When cf=1 coefficient=0 appears in only one category

When cf=2 coefficient=0.6 appears in two categories

When cf=3 coefficient=0.8 appears in three categories

Let us find out why my solution is more reliable. Suppose an unpopular movie has got 30 from rating and comment system, if the keyword appears only once, then final score is 30. If the keyword appears twice, the score will be 48, and if the keyword appears three times, the final score will be 54! Thus an unpopular movie will show up when the time is right.

4.2.3 term frequency

When we need to test the keyword in movie description, the term frequency is a powerful algorithm to find which movie is more related with the keyword.

1. Term Frequency Weight

The log frequency weight of term t in d is defined as follows

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d} & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

2. Idf Weight

The document frequency df_t is defined as the number of documents that t occurs in. We define the idf weight of term t as follows:

$$idf_t = \log_{10} \frac{N}{df_t}$$

3. Tf-idf Weight

The tf-idf weight of a term is the product of its tf weight and its idf weight:

$$w_{t,d} = (1 + \log tf_{t,d}) \cdot \log \frac{N}{df_t}$$

In my database, there are three kinds of term frequency: ABC, BC and C. Each one has different utilities. The C term frequency only comes from movie description and others' range are wider.

With term frequency, we can easily find the certain movies that contains the keyword many times.

5. Framework

Here are my four general steps to build DMASE:

1. Crawler
2. Extract data from the web page
3. Build index
4. Serving the service

1. Crawler

A crawler is also called a spider which is a program that is capable of iteratively and automatically extract data from website.

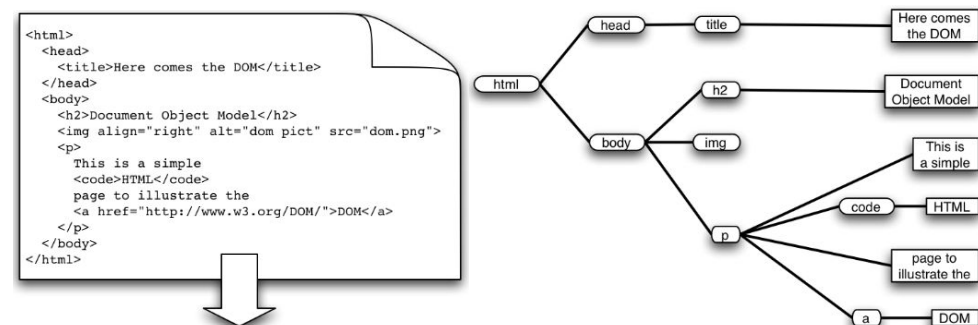
1. Begin with "seeds" URLs
2. Fetch and parse html body
3. Extract new URLs and find where they point to
4. Place the extracted URLs on a queue
5. Fetch each URL on the queue and repeat
6. Storing the scraped item

Given that python Scrapy framework is easily extensible, and portable, I chose Scrapy to build my spider .

2. Extract data from the web page

To extract data from the HTML, I have many tools: BeautifulSoup, lxml, etc.

To select nodes in XML, I chose XPath and CSS for they are easy to learn and easy to use.



3. Build index

Before index, I must do word segmentation. Here lists some of my alternative choices:

Chinese word segmentation:

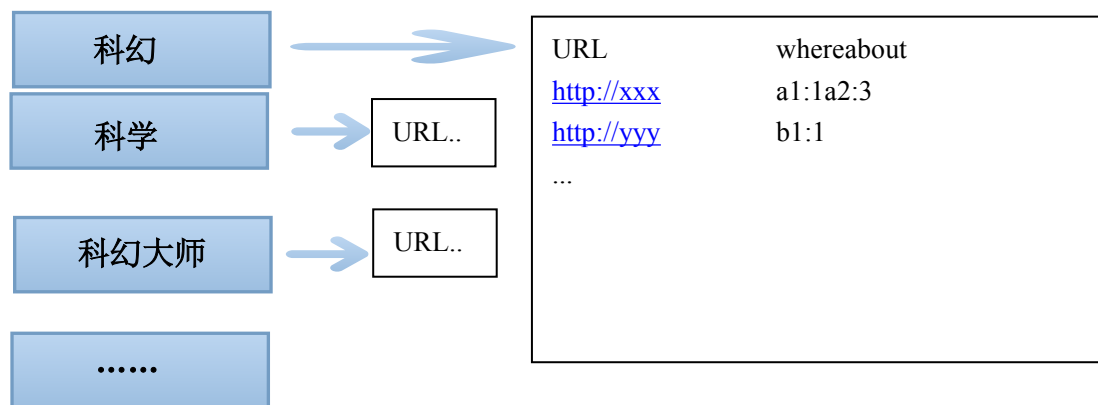
Jieba
NLPIR

Although NLPIR is more powerful, I still choose Jieba to build DMASE. The reason is I love Jieba' s elegance, delicacy and simplicity.

Index:

Lucene
Xapian
Sphinx

Well, I did not use any of them. I build my index myself. The structure of mine is:



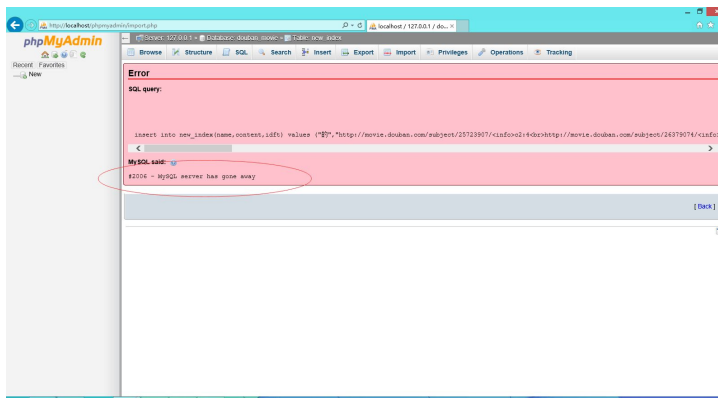
I stored my index and movie information in the JSON format.

4. Serving the service

The Database I chose is MySQL. The connection between MySQL and python is implemented by MySQLdb.

6. Details

6.1 thirty thousand is different from three thousand: the problem happens when data is huge



6.2 back slash in mySQL

```
def processSigns(string):
    stringSplit = string.split('\\')
    s = u''
    for littleS in stringSplit:
        s += littleS + r'\\'
    string = s[:-2]
    stringSplit = string.split('"')
    s = u''
    for littleS in stringSplit:
        s += littleS + r'\''
    s = s[:-2]
    return s
```

6.3 problems in movie titles

哈利波特 1: 神秘的魔法石(港/台) / 哈 1 / Harry Potter and the Philosopher's Stone
"仙履奇缘(港", "台")

6.4 uncommon signs

id	name	content	abc_dft	bc_dft	c_dft
259275		5967821<info>b1-1 5967821<info>b1-1	2	2	0
224256		3258585<info>c2-1 2060181<info>c2-2 1859528<...	9	9	9
198197		3258585<info>c2-1 2060181<info>c2-2 1859528<...	9	9	9
63707		1304201<info>c2-2 1304201<info>c2-2	2	2	2
38866		1304201<info>c2-2 3101546<info>c2-10 3101546<...	4	4	4
455895		3001600<info>c2-1	1	1	1
245959		1418903<info>c2-1 3861912<info>c2-1 1418903<...	3	3	3
404645		6508774<info>c2-1 1305281<info>c2-1 3861912<...	5	5	5
319579		3861912<info>c2-3	1	1	1
269301		2242338<info>b2-1 4818995<info>c2-8	2	2	1
244654		3897842<info>c2-12 3861912<info>c2-8 1866643<...	5	5	5
422142		3861912<info>c2-54 3101546<info>c2-19 3993335<...	6	6	6
106327		1304201<info>c2-1 3101546<info>c2-1 3993335<...	14	14	13
388090		1959296<info>c2-1 1959296<info>c2-1	2	2	2
271633		1959296<info>c2-3 1959296<info>c2-3	2	2	2
312611		5968334<info>c2-1 26206746<info>c2-1 2620674...	25	25	8
27678		3401751<info>b1-1 3401751<info>b1-1	2	2	0
4412	李胜永	3012307<info>a7-1	1	0	0
86765	李	3012307<info>b2-1	1	1	0
428025	游	24532162<info>c2-1	1	1	1
411480	丹	2132458<info>b1-1	1	1	0
326361		4910179<info>c2-6 4910179<info>c2-6 4910178<...	3	3	3

6.5 half-pitch and full-pitch!!

[illegible]

movie_information

Column	Type	Null	Default	Comments	MIME
id	int(11)	No			
title	varchar(1024)	No			
alias	text	No			
year	char(10)	Yes	NULL		

director	varchar(512)	Yes	<i>NULL</i>		
rating	char(10)	Yes	<i>NULL</i>		
comment_amount	int(11)	No			
betterthan	text	No			
intro	text	Yes	<i>NULL</i>		
link	char(100)	No			
writer	text	Yes	<i>NULL</i>		
actor	text	Yes	<i>NULL</i>		
info	text	Yes	<i>NULL</i>		

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	31408	A	No	
link	BTREE	Yes	No	link	31408	A	No	
link_search	BTREE	No	No	link		A	No	
rating_search	BTREE	No	No	rating		A	Yes	

new_index

Column	Type	Null	Default	Comments	MIME
id	int(11)	No			
name	char(35)	No			
content	mediumtext	No			
abc_dft	int(6)	No			
bc_dft	int(6)	No			
c_dft	int(6)	No			

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	459442	A	No	
name	BTREE	Yes	No	name	459442	A	No	
name_search	BTREE	No	No	name		A	No	

Index:

3	美剧生活	12.7889	http://movie.douban.com/subject/1483239/<info>c2
4	波兰	12.7889	http://movie.douban.com/subject/6721670/<info>c2/c2
5	RollsRoyce	0.0000	http://movie.douban.com/subject/5044466/<info>a3
6	跑步	12.7889	http://movie.douban.com/subject/10502527/<info>c2
7	Marie-Anne Fliegel	0.0000	http://movie.douban.com/subject/2213597/<info>a7
8	Jamie Oliver	0.0000	http://movie.douban.com/subject/2269936/<info>a7
9	小猫	12.7889	http://movie.douban.com/subject/3153640/<info>c2
10	未作	11.2137	http://movie.douban.com/subject/5383525/<info>c2 http://movie.douban.com/subject/4312428/<info>c2 http://movie.douban.com/subject/1899664/<info>c2
11	匿名	8.9915	http://movie.douban.com/subject/11601131/<info>c2 http://movie.douban.com/subject/8875610/<info>c2 http://movie.douban.com/subject/4845728/<info>c2 http://movie.douban.com/subject/1418680/<info>c2
12	120分钟(中国大陆)	0.0000	http://movie.douban.com/subject/10486467/<info>b3
13	hanging	12.7889	http://movie.douban.com/subject/3182807/<info>c2
14	电视新闻	10.7985	http://movie.douban.com/subject/3835489/<info>c2 http://movie.douban.com/subject/25830985/<info>c2 http://movie.douban.com/subject/3581466/<info>c2 http://movie.douban.com/subject/4944009/<info>c2

Movie information:

```

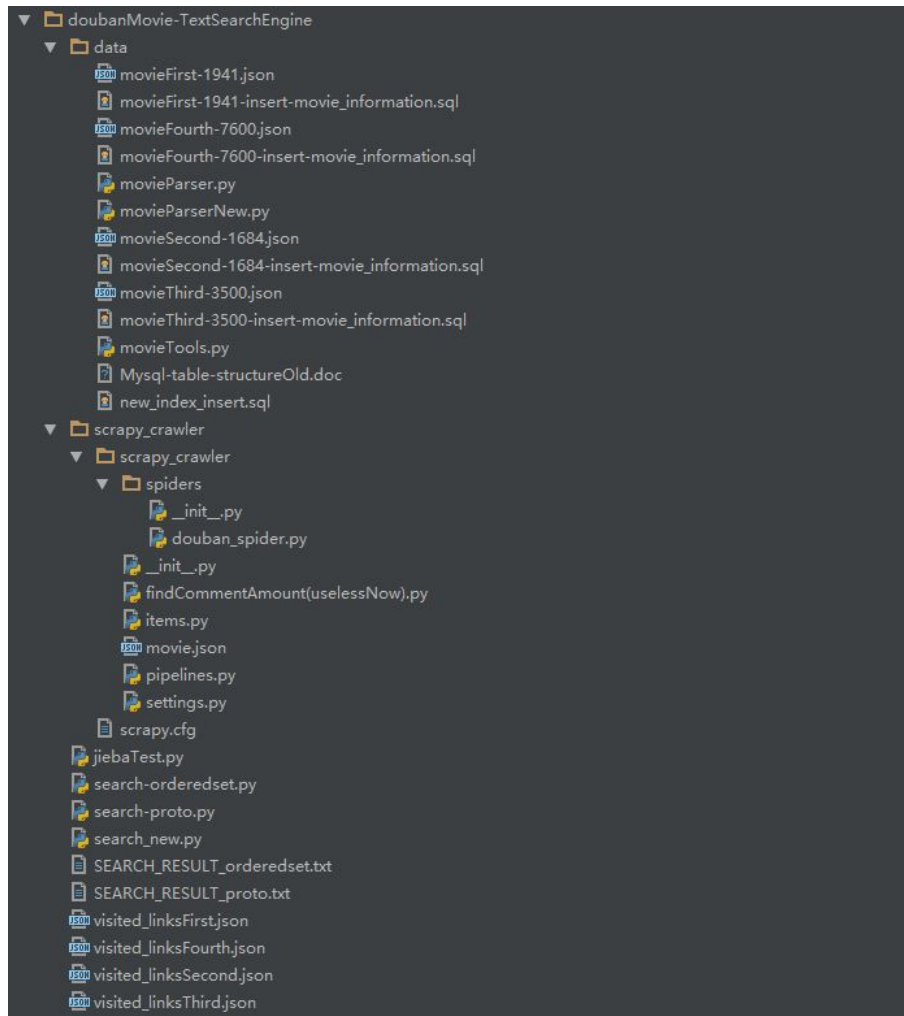
movieThird-3500.json - Microsoft Visual Studio
FILE EDIT VIEW PROJECT DEBUG TEAM TOOLS TEST ANALYZE WINDOW HELP
Quick Launch (Ctrl+Q)
Attach...
Frank He

movieThird-3500.json
<No Schema Selected>

[{"A_movieTitle": ["教父 The Godfather"], "B_movieYear": ["(1972)"], "C_movieDirector": ["弗朗西斯·福特·科波拉"], "D_movieRating": ["9.2"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["十二怒汉 12 Angry Men"], "B_movieYear": ["(1957)"], "C_movieDirector": ["西德尼·吕美特"], "D_movieRating": ["9.3"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["星球大战 Star Wars"], "B_movieYear": ["(1977)"], "C_movieDirector": ["乔治·卢卡斯"], "D_movieRating": ["8.3"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["教父2 The Godfather: Part II"], "B_movieYear": ["(1974)"], "C_movieDirector": ["弗朗西斯·福特·科波拉"], "D_movieRating": ["9.2"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["美国往事 Once Upon a Time in America"], "B_movieYear": ["(1984)"], "C_movieDirector": ["赛尔乔·莱翁内"], "D_movieRating": ["8.9"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["搏击俱乐部 Fight Club"], "B_movieYear": ["(1999)"], "C_movieDirector": ["大卫·芬奇"], "D_movieRating": ["9.0"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["英雄本色"], "B_movieYear": ["(1986)"], "C_movieDirector": ["吴宇森"], "D_movieRating": ["8.7"], "E_movieIntro": ["香港某个"], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["霸王别姬"], "B_movieYear": ["(1993)"], "C_movieDirector": ["陈凯歌"], "D_movieRating": ["9.4"], "E_movieIntro": ["段小楼 ("], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["低俗小说 Pulp Fiction"], "B_movieYear": ["(1994)"], "C_movieDirector": ["昆汀·塔伦蒂诺"], "D_movieRating": ["8.7"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["飞越疯人院 One Flew Over the Cuckoo's Nest"], "B_movieYear": ["(1975)"], "C_movieDirector": ["米洛斯·福尔曼"], "D_movieRating": ["8.9"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["窃听风暴 Das Leben der Anderen"], "B_movieYear": ["(2006)"], "C_movieDirector": ["弗洛里安·亨克尔·冯·多纳斯马"], "D_movieRating": ["8.8"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["控方证人 Witness for the Prosecution"], "B_movieYear": ["(1957)"], "C_movieDirector": ["比利·怀德"], "D_movieRating": ["9.0"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["楚门的世界 The Truman Show"], "B_movieYear": ["(1998)"], "C_movieDirector": ["彼得·威尔"], "D_movieRating": ["8.9"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["罗生门 羅生門"], "B_movieYear": ["(1950)"], "C_movieDirector": ["黑泽明"], "D_movieRating": ["8.7"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["辛德勒的名单 Schindler's List"], "B_movieYear": ["(1993)"], "C_movieDirector": ["史蒂文·斯皮尔伯格"], "D_movieRating": ["9.0"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["闻香识女人 Scent of a Woman"], "B_movieYear": ["(1992)"], "C_movieDirector": ["马丁·布莱斯"], "D_movieRating": ["8.9"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["星球大战2：帝国反击战 Star Wars: Episode V - The Empire Strikes Back"], "B_movieYear": ["(1980)"], "C_movieDirector": ["乔治·卢卡斯"], "D_movieRating": ["8.8"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["异形 Alien"], "B_movieYear": ["(1979)"], "C_movieDirector": ["雷德利·斯科特"], "D_movieRating": ["7.8"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["人猿星球 Planet of the Apes"], "B_movieYear": ["(1968)"], "C_movieDirector": ["富兰克林·沙夫纳"], "D_movieRating": ["8.0"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["机器战警 RoboCop"], "B_movieYear": ["(1987)"], "C_movieDirector": ["保罗·范霍文"], "D_movieRating": ["7.5"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["侏罗纪公园 Jurassic Park"], "B_movieYear": ["(1993)"], "C_movieDirector": ["史蒂文·斯皮尔伯格"], "D_movieRating": ["7.9"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["星球大战：克隆人战争 第一季 Star Wars: The Clone Wars Season 1 Season 1"], "B_movieYear": ["(2008)"], "C_movieDirector": ["乔治·卢卡斯"], "D_movieRating": ["7.8"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["非常嫌疑犯 The Usual Suspects"], "B_movieYear": ["(1995)"], "C_movieDirector": ["布赖恩·科佩曼斯基"], "D_movieRating": ["8.0"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["穆赫兰道 Mulholland Dr."], "B_movieYear": ["(2001)"], "C_movieDirector": ["大卫·林奇"], "D_movieRating": ["8.0"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["沉默的羔羊 The Silence of the Lambs"], "B_movieYear": ["(1991)"], "C_movieDirector": ["乔纳森·戴米"], "D_movieRating": ["8.6"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["上帝之城 Cidade de Deus"], "B_movieYear": ["(2002)"], "C_movieDirector": ["卡迪亚·兰德"], "D_movieRating": ["8.4"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["英国病人 The English Patient"], "B_movieYear": ["(1996)"], "C_movieDirector": ["安东尼·明格拉"], "D_movieRating": ["8.4"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}, {"A_movieTitle": ["卡萨布兰卡 Casablanca"], "B_movieYear": ["(1942)"], "C_movieDirector": ["迈克尔·柯蒂斯"], "D_movieRating": ["8.6"], "E_movieIntro": [""], "F_movieDirector": [""], "G_movieRating": [""], "H_movieIntro": [""], "I_movieRating": [""], "J_movieIntro": [""]}

```

Project:



7. Expectations

1. Words filter parser.

Some words like “йцук”, “ðšè” will never be searched by Chinese and English users, thus storing those words is waste of space.

2. Actor and director evaluation system

To make a good ranking system, actor and director evaluation system is very helpful.

3. Use a movie to search the movies

Given a already known movie, we can search movies by calculate the similarity between two movies.

4. Word expansion. Word correction.

5. Multi-cache or memory.

6. Multi-thread web server.