浪潮新闻文本分类及推荐

文本分类：

Text Classification with CNN and RNN

参考：https://github.com/gaussic/text-classification-cnn-rnn

环境

- Python 3
- TensorFlow 1.3
- numpy
- scikit-learn
- scipy

数据集：

使用 THUCNews 的一个子集进行训练与测试：http://thuctc.thunlp.org/

本次训练使用了其中的 10 个分类，每个分类 6500 条数据。

类别如下：

```
体育，财经，房产，家居，教育，科技，时尚，时政，游戏，娱乐
```

数据集划分如下：

- 训练集: 5000*10
- 验证集: 500*10
- 测试集: 1000*10

从原数据集生成子集的过程请参看 `helper` 下的两个脚本。其中，`copy_data.sh` 用于从每个分类拷贝 6500 个文件，`cnews_group.py` 用于将多个文件整合到一个文件中。执行该文件后，得到三个数据文件：

- cnews.train.txt: 训练集(50000 条)
- cnews.val.txt: 验证集(5000 条)
- cnews.test.txt: 测试集(10000 条)

预处理

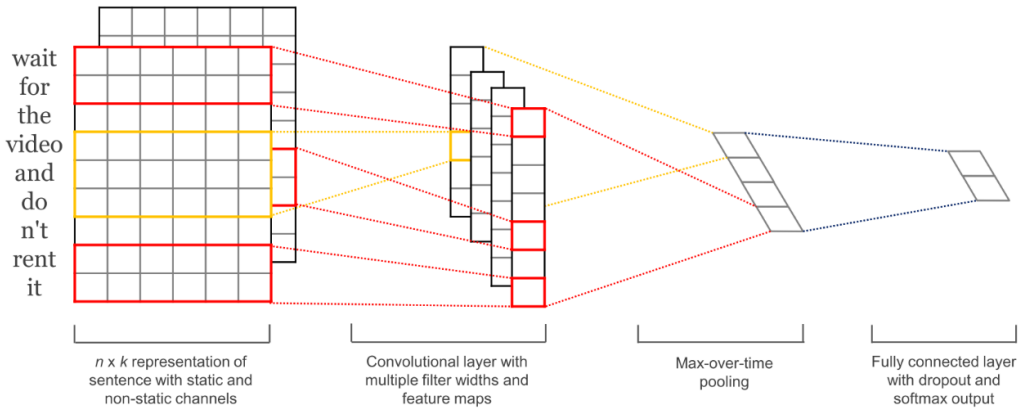
`data/cnews_loader.py` 为数据的预处理文件。
经过数据预处理，数据的格式如下：

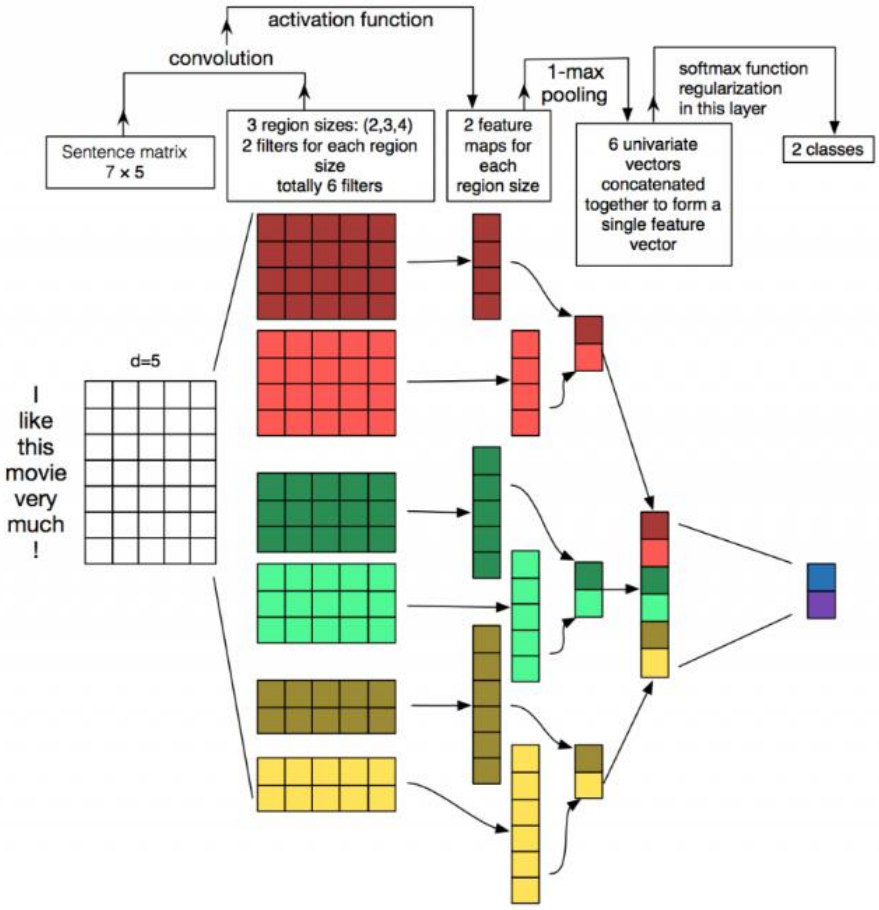
Data	Shape	Data	Shape
x_train	[50000, 600]	y_train	[50000, 10]
x_val	[5000, 600]	y_val	[5000, 10]
x_test	[10000, 600]	y_test	[10000, 10]

CNN 卷积神经网络

参考：https://blog.csdn.net/chuchus/article/details/77847476

TextCNN 是利用卷积神经网络对文本进行分类的算法，由 Yoon Kim 在 “Convolutional Neural Networks for Sentence Classification” 一文中提出. 是 2014 年的算法.





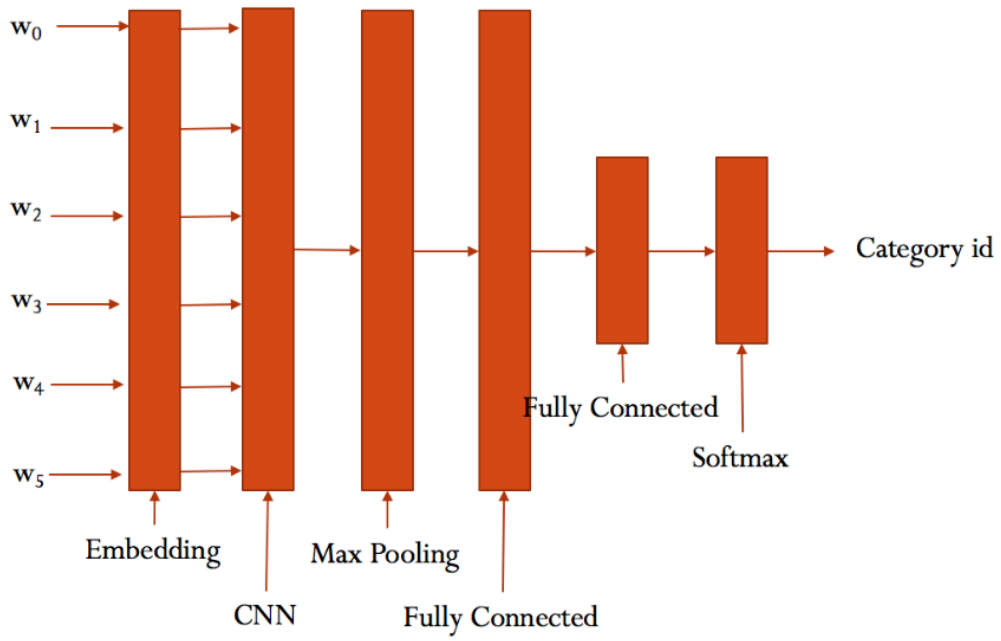
配置项

CNN 可配置的参数，在 cnn_model.py 中。

CNN 模型

具体参看 cnn_model.py 的实现。

大致结构如下：



训练与验证

运行 python run_cnn.py train，可以开始训练。

```
2018-05-31 15:49:08.578903: I tensorflow/core/common_runtime/gpu/gpu_device.cc:976] DMA: 0
2018-05-31 15:49:08.578910: I tensorflow/core/common_runtime/gpu/gpu_device.cc:986] 0:   Y
2018-05-31 15:49:08.578918: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1045] Creating TensorFlow d
Training and evaluating...
Epoch: 1
Iter:      0, Train Loss:    2.3, Train Acc:  10.94%, Val Loss:    2.3, Val Acc:  10.00%, Time: 0:00:04 *
Iter:    100, Train Loss:    1.2, Train Acc:  62.50%, Val Loss:    1.1, Val Acc:  69.10%, Time: 0:00:09 *
Iter:    200, Train Loss:   0.45, Train Acc:  84.38%, Val Loss:   0.65, Val Acc:  78.48%, Time: 0:00:13 *
Iter:    300, Train Loss:   0.25, Train Acc:  93.75%, Val Loss:   0.46, Val Acc:  85.68%, Time: 0:00:17 *
Iter:    400, Train Loss:   0.36, Train Acc:  90.62%, Val Loss:   0.35, Val Acc:  89.58%, Time: 0:00:22 *
Iter:    500, Train Loss:    0.2, Train Acc:  96.88%, Val Loss:   0.32, Val Acc:  89.66%, Time: 0:00:26 *
Iter:    600, Train Loss:   0.11, Train Acc:  96.88%, Val Loss:   0.28, Val Acc:  92.14%, Time: 0:00:30 *
Iter:    700, Train Loss:   0.07, Train Acc:  95.31%, Val Loss:   0.25, Val Acc:  92.94%, Time: 0:00:34 *
Epoch: 2
Iter:    800, Train Loss:  0.091, Train Acc:  96.88%, Val Loss:   0.23, Val Acc:  93.56%, Time: 0:00:39 *
Iter:    900, Train Loss:  0.079, Train Acc:  98.44%, Val Loss:   0.24, Val Acc:  93.48%, Time: 0:00:42
Iter:   1000, Train Loss:   0.14, Train Acc:  96.88%, Val Loss:   0.23, Val Acc:  93.76%, Time: 0:00:46 *
Iter:   1100, Train Loss:   0.21, Train Acc:  96.88%, Val Loss:   0.22, Val Acc:  94.10%, Time: 0:00:51 *
Iter:   1200, Train Loss:  0.081, Train Acc:  98.44%, Val Loss:   0.21, Val Acc:  94.60%, Time: 0:00:55 *
Iter:   1300, Train Loss:  0.097, Train Acc:  96.88%, Val Loss:   0.22, Val Acc:  94.06%, Time: 0:00:59
Iter:   1400, Train Loss:   0.14, Train Acc:  98.44%, Val Loss:   0.23, Val Acc:  93.74%, Time: 0:01:04
Iter:   1500, Train Loss:  0.084, Train Acc:  98.44%, Val Loss:   0.23, Val Acc:  93.24%, Time: 0:01:08
Epoch: 3
Iter:   1600, Train Loss:  0.091, Train Acc:  96.88%, Val Loss:   0.23, Val Acc:  94.16%, Time: 0:01:11
Iter:   1700, Train Loss:  0.027, Train Acc: 100.00%, Val Loss:   0.21, Val Acc:  94.62%, Time: 0:01:16 *
Iter:   1800, Train Loss:   0.18, Train Acc:  93.75%, Val Loss:   0.19, Val Acc:  95.16%, Time: 0:01:20 *
Iter:   1900, Train Loss:  0.083, Train Acc:  98.44%, Val Loss:   0.21, Val Acc:  93.60%, Time: 0:01:25
Iter:   2000, Train Loss:  0.027, Train Acc:  98.44%, Val Loss:    0.2, Val Acc:  94.76%, Time: 0:01:29
Iter:   2100, Train Loss:   0.15, Train Acc:  96.88%, Val Loss:    0.2, Val Acc:  94.94%, Time: 0:01:32
Iter:   2200, Train Loss:   0.23, Train Acc:  95.31%, Val Loss:   0.21, Val Acc:  94.54%, Time: 0:01:34
Iter:   2300, Train Loss:  0.051, Train Acc:  98.44%, Val Loss:    0.2, Val Acc:  94.92%, Time: 0:01:36
Epoch: 4
Iter:   2400, Train Loss:  0.034, Train Acc:  98.44%, Val Loss:   0.21, Val Acc:  94.90%, Time: 0:01:38
Iter:   2500, Train Loss:  0.032, Train Acc: 100.00%, Val Loss:   0.22, Val Acc:  93.86%, Time: 0:01:40
Iter:   2600, Train Loss:  0.058, Train Acc:  98.44%, Val Loss:    0.2, Val Acc:  94.80%, Time: 0:01:41
Iter:   2700, Train Loss:  0.026, Train Acc: 100.00%, Val Loss:   0.21, Val Acc:  94.36%, Time: 0:01:43
Iter:   2800, Train Loss:  0.085, Train Acc:  96.88%, Val Loss:   0.23, Val Acc:  93.78%, Time: 0:01:45
No optimization for a long time, auto-stopping...
embedded@embedded:~/news/text-classification-cnn-rnn-master/text-classification-cnn-rnn-master$
```

测试

运行 python run_cnn.py test 在测试集上进行测试。

```
Testing...
Test Loss:   0.13, Test Acc:  95.99%
Precision, Recall and F1-Score...
             precision    recall  f1-score   support

         体育       1.00      0.99      0.99      1000
         财经       0.97      0.99      0.98      1000
         房产       1.00      1.00      1.00      1000
         家居       0.97      0.85      0.91      1000
         教育       0.90      0.94      0.92      1000
         科技       0.94      0.98      0.96      1000
         时尚       0.91      0.98      0.95      1000
         时政       0.96      0.92      0.94      1000
         游戏       0.99      0.97      0.98      1000
         娱乐       0.97      0.98      0.97      1000

avg / total       0.96      0.96      0.96     10000

Confusion Matrix...
[[994   0   0   0   3   1   0   2   0   0]
 [  0 991   0   0   3   1   0   5   0   0]
 [  0   1 997   1   1   0   0   0   0   0]
 [  1  11   0 854  26  18  53  26   1  10]
 [  1   6   0   6 936  17  16   5   4   9]
 [  0   0   0   4   4 979   9   0   3   1]
 [  1   0   0   2   7   3 980   0   1   6]
 [  0  14   0   4  40  13   0 924   1   4]
 [  0   2   0   1   9   4  11   0 968   5]
 [  1   0   0   4   8   6   4   1   0 976]]
Time usage: 0:00:07
```

RNN 循环神经网络

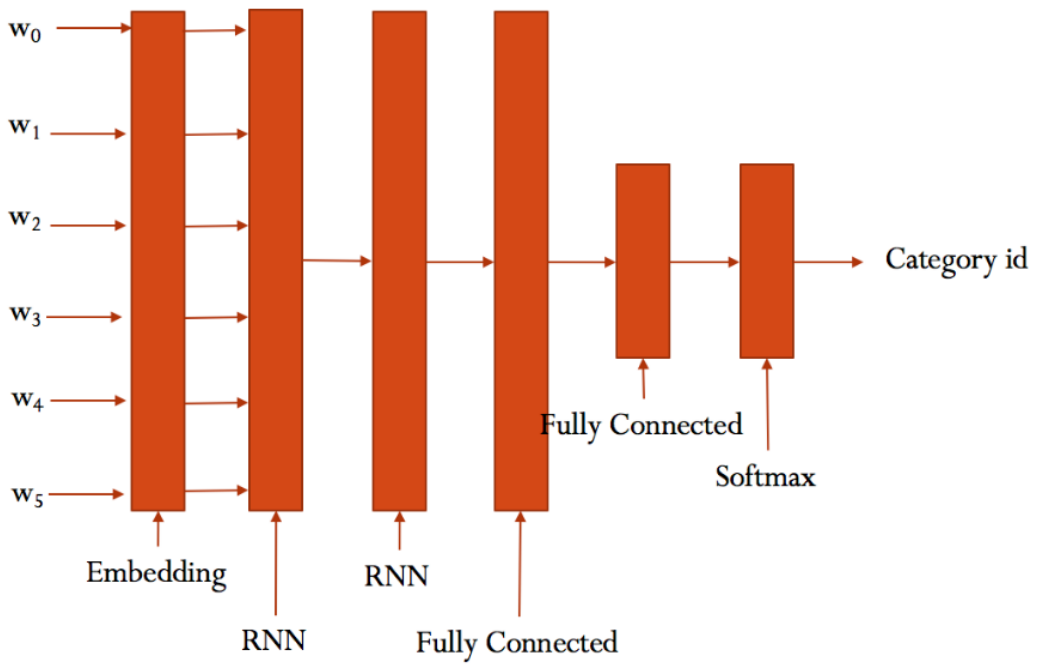
配置项

RNN 可配置的参数，在 rnn_model.py 中。

RNN 模型

具体参看 rnn_model.py 的实现。

大致结构如下：



训练与验证

运行 `python run_rnn.py train`，可以开始训练。

```
Iter:    300, Train Loss:   0.46, Train Acc:  85.16%, Val Loss:   0.51, Val Acc:  86.58%, Time: 0:05:18 *
Epoch: 2
Iter:    400, Train Loss:   0.18, Train Acc:  93.75%, Val Loss:   0.48, Val Acc:  86.98%, Time: 0:07:00 *
Iter:    500, Train Loss:   0.19, Train Acc:  92.97%, Val Loss:   0.45, Val Acc:  88.08%, Time: 0:08:42 *
Iter:    600, Train Loss:    0.2, Train Acc:  96.09%, Val Loss:    0.4, Val Acc:  89.42%, Time: 0:10:25 *
Iter:    700, Train Loss:   0.16, Train Acc:  94.53%, Val Loss:    0.5, Val Acc:  86.58%, Time: 0:12:07
Epoch: 3
Iter:    800, Train Loss:   0.25, Train Acc:  92.19%, Val Loss:   0.44, Val Acc:  89.66%, Time: 0:13:49 *
Iter:    900, Train Loss:   0.25, Train Acc:  92.97%, Val Loss:   0.47, Val Acc:  89.12%, Time: 0:15:31
Iter:   1000, Train Loss:    0.2, Train Acc:  93.75%, Val Loss:   0.55, Val Acc:  85.78%, Time: 0:17:13
Iter:   1100, Train Loss:  0.093, Train Acc:  97.66%, Val Loss:   0.51, Val Acc:  86.74%, Time: 0:18:55
Epoch: 4
Iter:   1200, Train Loss:   0.21, Train Acc:  95.31%, Val Loss:   0.36, Val Acc:  90.78%, Time: 0:20:37 *
Iter:   1300, Train Loss:   0.12, Train Acc:  94.53%, Val Loss:   0.41, Val Acc:  90.16%, Time: 0:22:18
Iter:   1400, Train Loss:   0.12, Train Acc:  96.88%, Val Loss:   0.45, Val Acc:  88.84%, Time: 0:24:00
Iter:   1500, Train Loss:  0.062, Train Acc:  97.66%, Val Loss:   0.39, Val Acc:  91.00%, Time: 0:25:42 *
Epoch: 5
Iter:   1600, Train Loss:   0.17, Train Acc:  96.88%, Val Loss:   0.36, Val Acc:  90.82%, Time: 0:27:24
Iter:   1700, Train Loss:  0.099, Train Acc:  96.09%, Val Loss:   0.34, Val Acc:  91.36%, Time: 0:29:07 *
Iter:   1800, Train Loss:   0.21, Train Acc:  95.31%, Val Loss:   0.42, Val Acc:  89.84%, Time: 0:30:47
Iter:   1900, Train Loss:   0.12, Train Acc:  97.66%, Val Loss:   0.31, Val Acc:  91.86%, Time: 0:32:30 *
Epoch: 6
Iter:   2000, Train Loss:  0.063, Train Acc:  97.66%, Val Loss:   0.43, Val Acc:  89.02%, Time: 0:34:11
Iter:   2100, Train Loss:  0.098, Train Acc:  96.88%, Val Loss:   0.39, Val Acc:  90.58%, Time: 0:35:53
Iter:   2200, Train Loss:    0.1, Train Acc:  96.88%, Val Loss:    0.4, Val Acc:  91.10%, Time: 0:37:35
Iter:   2300, Train Loss:   0.03, Train Acc:  99.22%, Val Loss:   0.39, Val Acc:  89.56%, Time: 0:39:18
Epoch: 7
Iter:   2400, Train Loss:   0.07, Train Acc:  97.66%, Val Loss:   0.32, Val Acc:  92.60%, Time: 0:41:01 *
Iter:   2500, Train Loss:  0.099, Train Acc:  98.44%, Val Loss:   0.36, Val Acc:  91.42%, Time: 0:42:44
Iter:   2600, Train Loss:  0.066, Train Acc:  98.44%, Val Loss:    0.3, Val Acc:  92.06%, Time: 0:44:27
Iter:   2700, Train Loss:  0.065, Train Acc:  97.66%, Val Loss:   0.35, Val Acc:  91.28%, Time: 0:46:10
Epoch: 8
Iter:   2800, Train Loss:  0.055, Train Acc:  97.66%, Val Loss:   0.42, Val Acc:  90.82%, Time: 0:47:53
Iter:   2900, Train Loss:   0.08, Train Acc:  98.44%, Val Loss:   0.35, Val Acc:  91.98%, Time: 0:49:34
Iter:   3000, Train Loss:   0.02, Train Acc:  99.22%, Val Loss:   0.36, Val Acc:  91.32%, Time: 0:51:16
Iter:   3100, Train Loss:   0.07, Train Acc:  96.88%, Val Loss:   0.36, Val Acc:  91.10%, Time: 0:52:58
Epoch: 9
Iter:   3200, Train Loss:    0.1, Train Acc:  97.66%, Val Loss:   0.35, Val Acc:  91.58%, Time: 0:54:40
Iter:   3300, Train Loss:  0.083, Train Acc:  98.44%, Val Loss:    0.4, Val Acc:  90.62%, Time: 0:56:24
Iter:   3400, Train Loss:  0.028, Train Acc:  99.22%, Val Loss:   0.41, Val Acc:  90.32%, Time: 0:58:07
No optimization for a long time, auto-stopping...
```

测试

运行 `python run_rnn.py test` 在测试集上进行测试。

```
Testing...
Test Loss:   0.18, Test Acc:  95.21%
Precision, Recall and F1-Score...
             precision    recall  f1-score   support

         体育       1.00      0.96      0.98      1000
         财经       0.97      0.98      0.97      1000
         房产       1.00      1.00      1.00      1000
         家居       0.95      0.87      0.91      1000
         教育       0.89      0.93      0.91      1000
         科技       0.95      0.98      0.96      1000
         时尚       0.93      0.97      0.95      1000
         时政       0.92      0.92      0.92      1000
         游戏       0.96      0.96      0.96      1000
         娱乐       0.95      0.96      0.96      1000

avg / total       0.95      0.95      0.95     10000

Confusion Matrix...
[[959   0   0   5   2   6   0   4  10  14]
 [  0 977   2   1   2   5   2  11   0   0]
 [  0   1 996   1   2   0   0   0   0   0]
 [  0  11   2 870  29  10  37  30   6   5]
 [  0   4   0   6 933  11   3  28   4  11]
 [  0   3   1   0   5 976   3   4   8   0]
 [  1   0   0  12  10   2 970   0   1   4]
 [  0  14   0   9  37  11   0 922   3   4]
 [  1   1   0   1  13   3  15   0 956  10]
 [  0   1   0   6  12   5  11   0   3 962]]
Time usage: 0:00:42
```

分类（预测）

为方便预测， `predict.py` 提供了 CNN 模型的预测方法。我们即采用 CNN 模型进行新闻文本分类。

运行 `python predict.py`

读取.xlsx 数据先对内容进行解码，然后将标题加内容一起利用模型进行分类（预测）。

以下是十个类别，和新闻分类结果：

id	name
0	体育
1	财经
2	房产
3	家居
4	教育
5	科技
6	时尚
7	时政
8	游戏
9	娱乐

id	module_id	title	content	news_time	browse_times
4862	4	济南普高推荐生选	给你一个酒精灯、一个石	43247.628275	461
4863	7	明府城西区总规划	明府城片区是济南的一块	43247.598519	694
4864	7	济南历下区前四个	27日，济南市2018年第	43247.483993	734
4865	5	济南鱼翅皇宫附近	鱼翅皇宫大酒店以及周边	43247.482951	862
4866	7	内蒙古毕拉河探索	“前些年，道路太差，往	43247.400255	540
4867	7	乌海市首批2个市级	经自治区标准化院标准审	43247.398252	428
4868	7	内蒙古乌海市政府	乌海市政府秘书长、党组	43247.39463	374
4869	7	乌海市地毯式排查	围绕中央环保督察反馈意	43247.390289	415
4870	5	宫颈癌疫苗，我能	<img src="http://icity2	43247.386146	659
4871	5	宫颈癌疫苗，我能	<img src="http://icity2	43247.384653	431
4872	7	赞！山东省将实行	记者从省发改委获悉，近	43247.377222	455
4873	4	山东省81所高校设	经前期审核汇总,日前省教	43247.374884	853
4874	5	前4月我省规模以上	记者近日从省统计局获悉	43247.372928	804
4875	4	2018就业率最高专	高考将至，在这之后，百	43247.343981	617
4876	5	孙丕恕出席2018中	5月26日上午，主题为“	43247.329653	641
4877	7	济南市政府领导班	济南市政府官方网站进行	43247.322824	1420
4878	7	济南首家社区戒毒	5月25日，山东省鲁中强	43247.316481	476
4879	8	刚刚，浪潮云又放	<div style="text-align:	43246.771655	1233
4880	7	国家发钱了、农民	当前部分地方存在不少真	43246.673056	792

新闻推荐：

说明：本推荐系统使用的推荐算法包括协同过滤（Collaborative Filtering）、基于内容相似度的推荐（Content-based Recommendation）与热点新闻推荐（Hot News Recommendation）：

- 协同过滤的实现依托于 Mahout 的提供库，即用户相似性；
- 基于内容的相似度推荐即内容相似进行推荐；

- 热点新闻推荐顾名思义是取最近被最多用户浏览过的新闻进行推荐

主要使用的库（Lib）：

- Ansj：基于内容的推荐部分用以分词，以及其内含的 TFIDF 算法。
- Quartz：推荐系统定时运行的设定。
- Mahout：使用内置的协同过滤算法。
- Jfinal：使用内置的 ActiveRecord 与 Db 工具，对推荐系统中的数据库表做了实体类映射，以简化数据库相关操作。

数据库

与 MYSQL 数据库进行交互

五个表：用户表（users），新闻表（news），新闻模块表(newsmodules)，浏览记录表（newslogs），推荐结果表（Recommendations）。

- 用户表 users

名	类型	长度	小数点	不是 null	
id	bigint	20	0	☑	🔑1
pref_list	text	0	0	☑	
latest_log_time	timestamp	0	0	☑	
name	varchar	255	0	☑	
password	varchar	255	0	☑	

- 新闻表 news

名	类型	长度	小数点	不是 null	
id	bigint	20	0	☑	🔑1
module_id	int	11	0	☐	
title	text	0	0	☐	
content	text	0	0	☐	
news_time	double	0	0	☐	
browse_times	bigint	255	0	☐	

- 新闻模块表 newsmodules

名	类型	长度	小数点	不是 null	
id	int	11	0	☑	🔑1
name	text	0	0	☑	

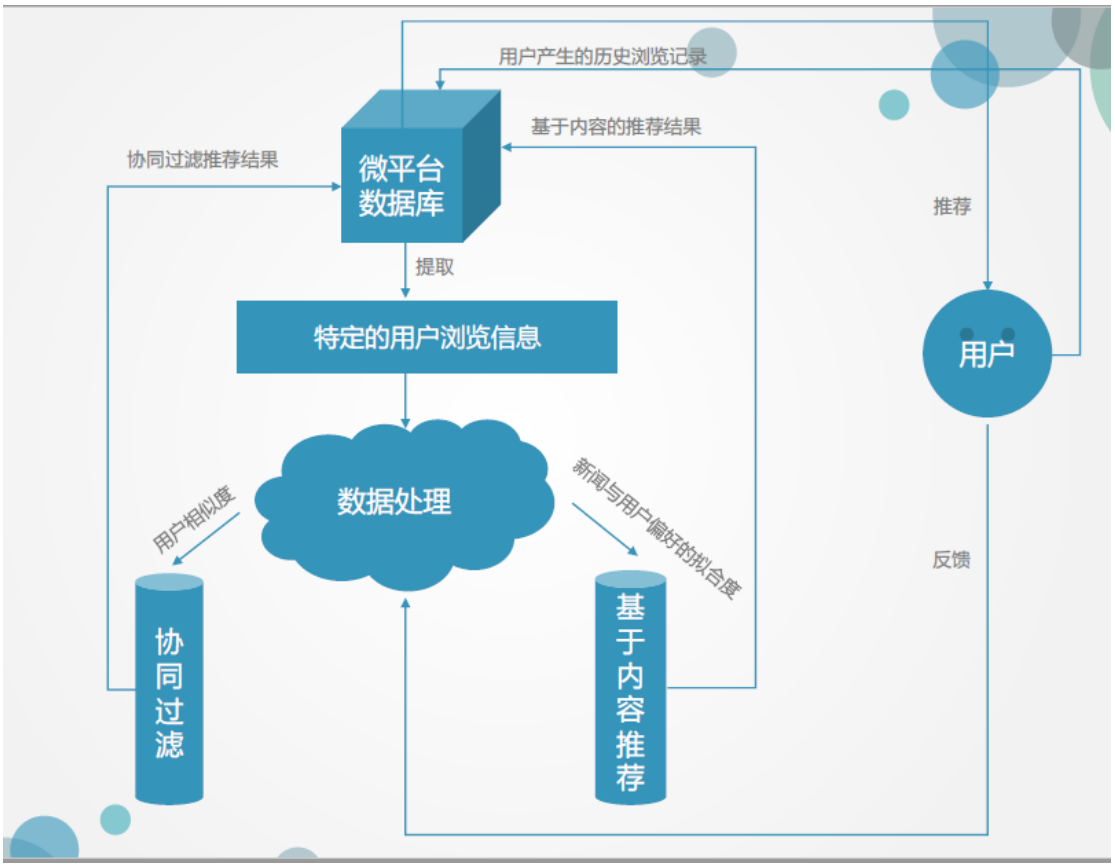
- 浏览记录表 newslogs

名	类型	长度	小数点	不是 null	
id	bigint	20	0	☑	🔑1
user_id	bigint	20	0	☑	
news_id	bigint	20	0	☑	
view_time	timestamp	0	0	☑	
prefer_degree	int	11	0	☑	

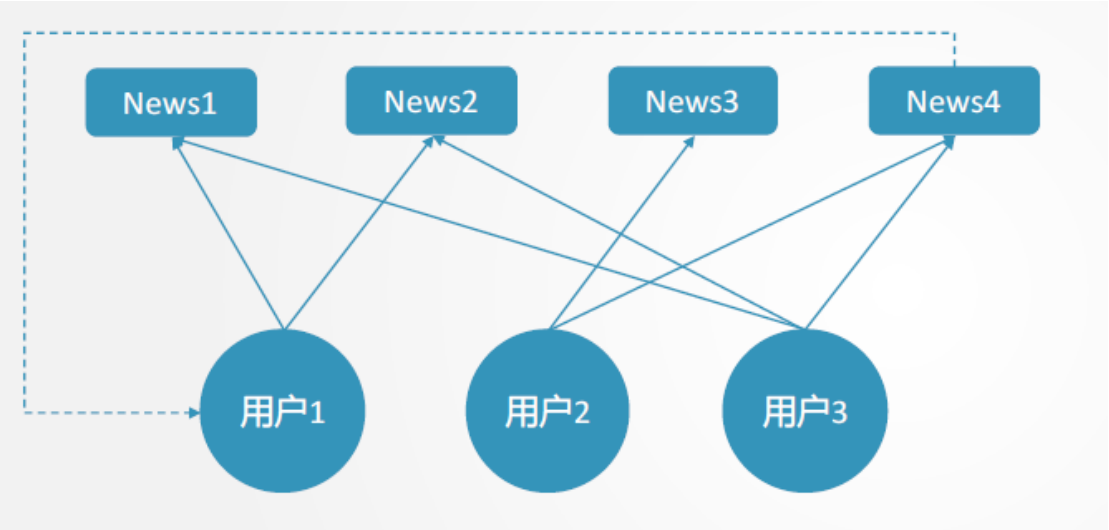
- 推荐结果表 Recommendations

名	类型	长度	小数点	不是 null	
id	bigint	20	0	☑	🔑1
user_id	bigint	20	0	☑	
news_id	bigint	20	0	☑	
derive_time	timestamp	0	0	☑	
feedback	bit	1	0	☐	
derive_algorithm	int	11	0	☑	

整体框架：

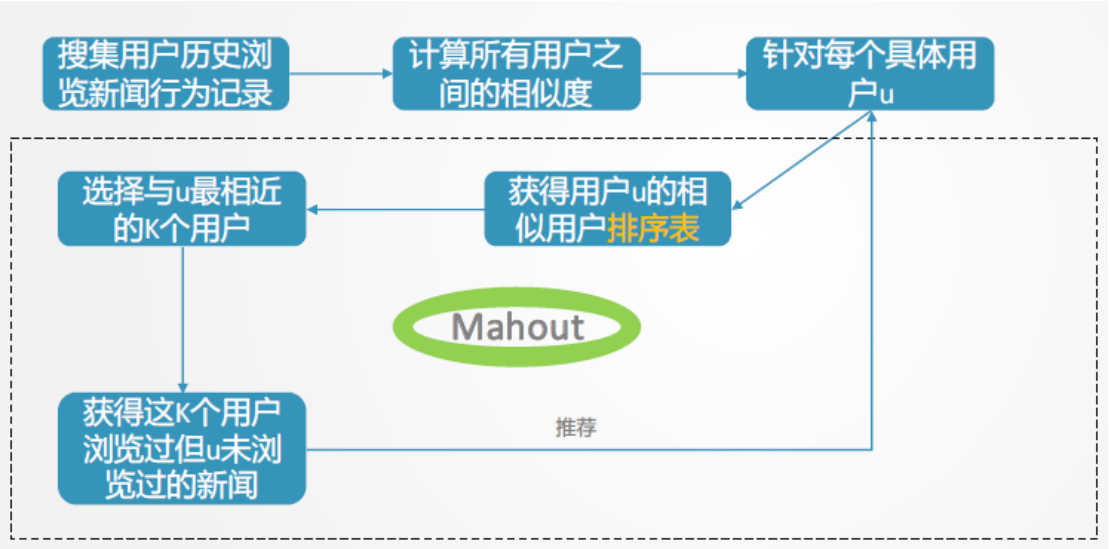


基于协同过滤的推荐：

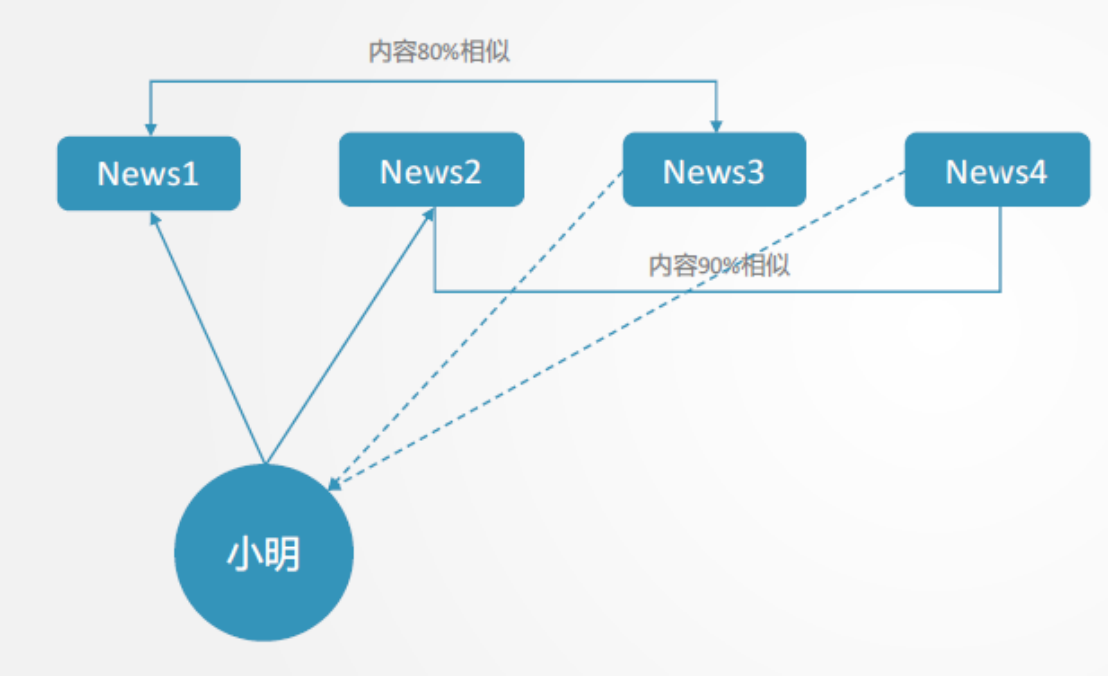


用户 1 和用户 3 的爱好更具有相似性，我们就把用户 3 看过但用户 1 未看过的新闻推荐给用户 1。

实现过程：



基于内容的推荐：



例如用户小明看过 1、2，但是通过对比发现，1 和 3 有 80%的内容相似，2 和 4 有 90%相似，于是，将 3 和 4 也推荐给用户小明。

用户喜好：

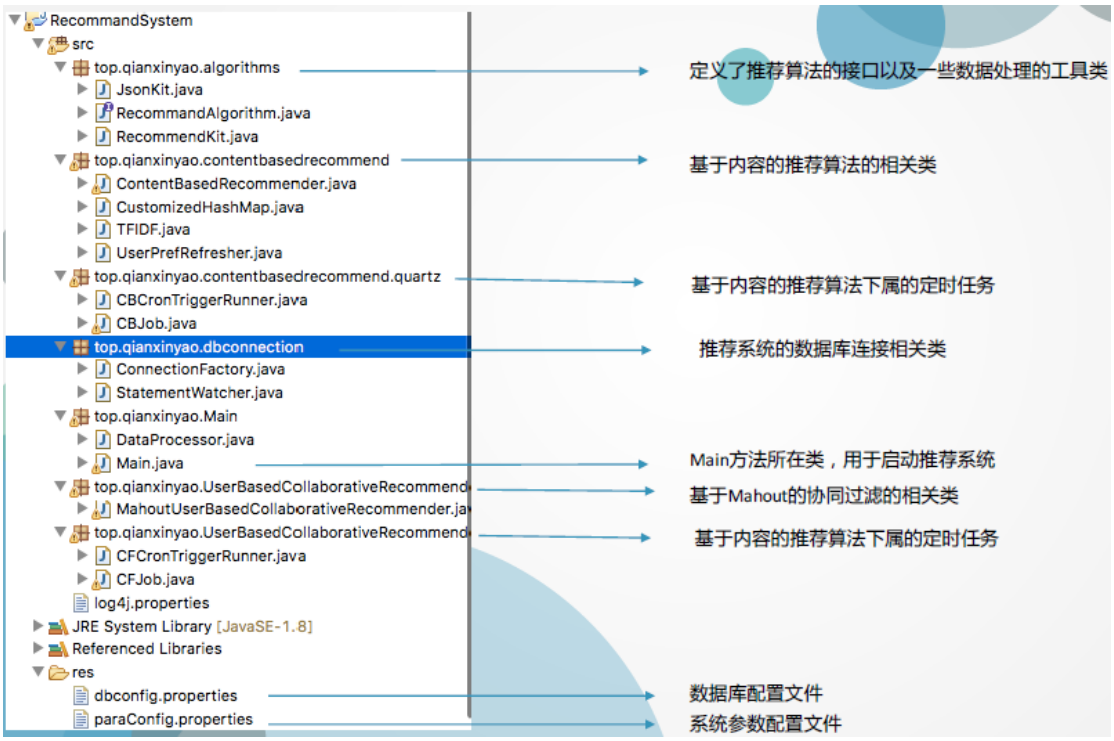
从用户历史浏览记录里利用 TF-IDF 算法（term frequency - inverse document frequency）挖掘用户喜好关键词，存入用户表的偏好。

TF-IDF 是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份

文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

为关键词列表设置一个衰减系数λ ，定期对用户的喜好关键词的 TF-IDF 值进行更新，减少关键词的收敛倾向。

工程结构：



初始用户浏览信息表：

id	user_id	news_id	view_time	prefer_degree
10	1	4872	2018-05-31 19:52:43	0
11	1	4873	2018-05-31 19:52:34	0
12	1	4874	2018-05-31 19:52:53	0
13	1	4875	2018-05-31 19:53:02	0
14	1	4876	2018-05-31 19:53:21	0
15	1	4877	2018-05-31 19:53:26	0
16	2	4872	2018-05-31 19:53:34	0
17	2	4873	2018-05-31 19:53:39	0
18	2	4875	2018-05-31 19:53:46	0
19	2	4877	2018-05-31 19:53:52	0
20	3	4873	2018-05-31 19:53:59	0
21	3	4874	2018-05-31 19:54:04	0
22	3	4875	2018-05-31 19:54:13	0
23	3	4876	2018-05-31 19:54:22	0
24	3	4877	2018-05-31 19:54:37	0

初始用户信息表：

id	pref_list	latest_log_time	name	password
1	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 21:01:42	aaa	123
2	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 21:01:39	bbb	123
3	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 21:01:37	ccc	123
4	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 20:59:59	ddd	123
5	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 21:00:01	eee	123
6	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 21:00:03	fff	123

完成用户喜好提取算法后：

id	pref_list	latest_log_time	name	password
1	{"0":{},"1":{},"2":{},"3":{},"4":{"教育厅":26.66297956059023,"教授":92.10344589152318,"科技厅":26.40598698651225,"技术":32.697887193502844,"高专":31.917	2018-05-31 21:02:34	aaa	123
2	{"0":{},"1":{},"2":{},"3":{},"4":{"教育厅":26.66297956059023,"教授":92.10344589152318,"科技厅":26.40598698651225,"技术":32.697887193502844,"高专":31.917	2018-05-31 21:02:34	bbb	123
3	{"0":{},"1":{},"2":{},"3":{},"4":{"教育厅":26.66297956059023,"教授":92.10344589152318,"科技厅":26.40598698651225,"技术":32.697887193502844,"高专":31.917	2018-05-31 21:02:34	ccc	123
4	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 20:59:59	ddd	123
5	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 21:00:01	eee	123
6	{"0":{},"1":{},"2":{},"3":{},"4":{},"5":{},"6":{},"7":{},"8":{},"9":{}}	2018-05-31 21:00:03	fff	123

完成新闻推荐算法后：

id	user_id	news_id	derive_time	feedback	derive_algorithm
43	2	4874	2018-05-31 20:45:56	(Null)	0
44	2	4876	2018-05-31 20:45:56	(Null)	0
45	3	4872	2018-05-31 20:45:56	(Null)	0
46	4	4872	2018-05-31 20:46:03	(Null)	2
47	4	4873	2018-05-31 20:46:03	(Null)	2
48	4	4874	2018-05-31 20:46:03	(Null)	2
49	4	4875	2018-05-31 20:46:03	(Null)	2
50	4	4876	2018-05-31 20:46:03	(Null)	2
51	4	4877	2018-05-31 20:46:03	(Null)	2
52	5	4872	2018-05-31 20:46:03	(Null)	2
53	5	4873	2018-05-31 20:46:03	(Null)	2
54	5	4874	2018-05-31 20:46:03	(Null)	2
55	5	4875	2018-05-31 20:46:03	(Null)	2
56	5	4876	2018-05-31 20:46:03	(Null)	2
57	5	4877	2018-05-31 20:46:03	(Null)	2
58	6	4872	2018-05-31 20:46:03	(Null)	2
59	6	4873	2018-05-31 20:46:03	(Null)	2
60	6	4874	2018-05-31 20:46:03	(Null)	2
61	6	4875	2018-05-31 20:46:03	(Null)	2
62	6	4876	2018-05-31 20:46:03	(Null)	2
63	6	4877	2018-05-31 20:46:03	(Null)	2

效果展示：（待开发中）

用户登录：

新闻阅读：

第19期

济南普高推荐生选拔进行中，山师附中考“煮鹌鹑蛋”

2018-05-31 09:02:10　我有话说(0人参与)

导读　给你一个酒精灯、一个石棉网、一个三脚架、一张A4纸、铁丝少许、胶带、打火机一个、钳子一把、剪刀一把、矿泉水一瓶，你如何将“生”的鹌鹑蛋，变成“熟”的鹌鹑蛋呢？

这不是蓝翔技校的烹饪班，而是山东师范大学附属中学的推荐生考试“科学素质、实践能力和发展潜能”部分的考题，现在考试正在进行中。

记者在考场外看到，同学们三人一组，正在想方设法煮蛋中。

评论

分享

关闭