Weekly report

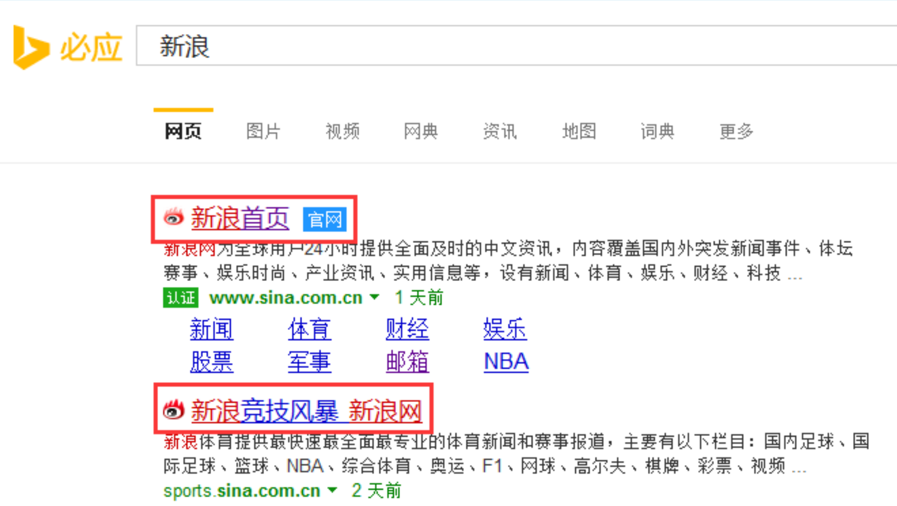
Tianyi Luo

Plan last two weeks

- Write a crawler based on keywords
- Utilize learning to rank technology to get best QA re-ranking results

Write a crawler based on keywords

- What problem we need to solve?
  - Given a list of key words which contains Chinese or English e.g. 新浪 阿里巴巴 Tencent.
  - We want to crawl the webpages returned by search engine e.g. Baidu and Bing using these keywords as query.



新浪首页 官网

新浪网为全球用户24小时提供全面及时的中文资讯，内容覆盖国内外突发新闻事件、体坛赛事、娱乐时尚、产业资讯、实用信息等，设有新闻、体育、娱乐、财经、科技 ...

认证 www.sina.com.cn ▾ 1 天前

新闻 体育 财经 娱乐
股票 军事 邮箱 NBA

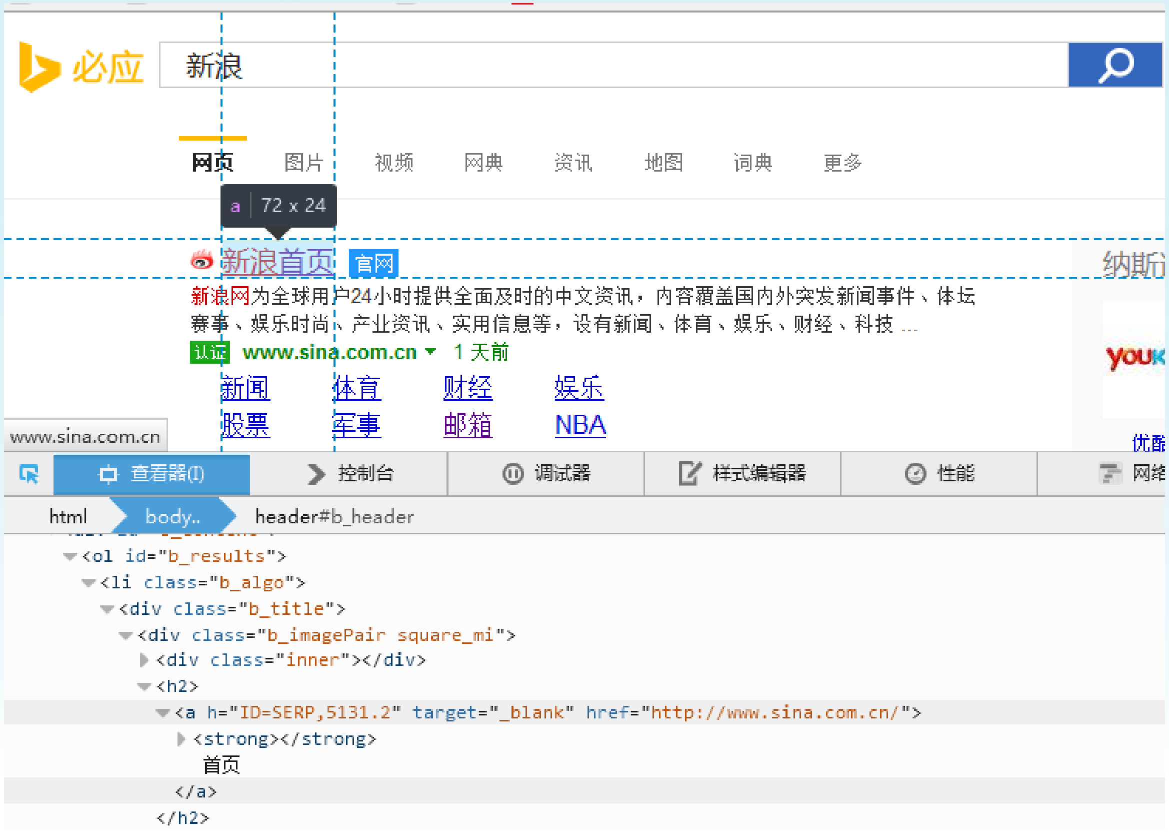
新浪竞技风暴_新浪网

新浪体育提供最快速最全面最专业的体育新闻和赛事报道，主要有以下栏目：国内足球、国际足球、篮球、NBA、综合体育、奥运、F1、网球、高尔夫、棋牌、彩票、视频 ...

sports.sina.com.cn ▾ 2 天前

Write a crawler based on keywords

- How we solve the link extraction problem?
  - Regular Expression



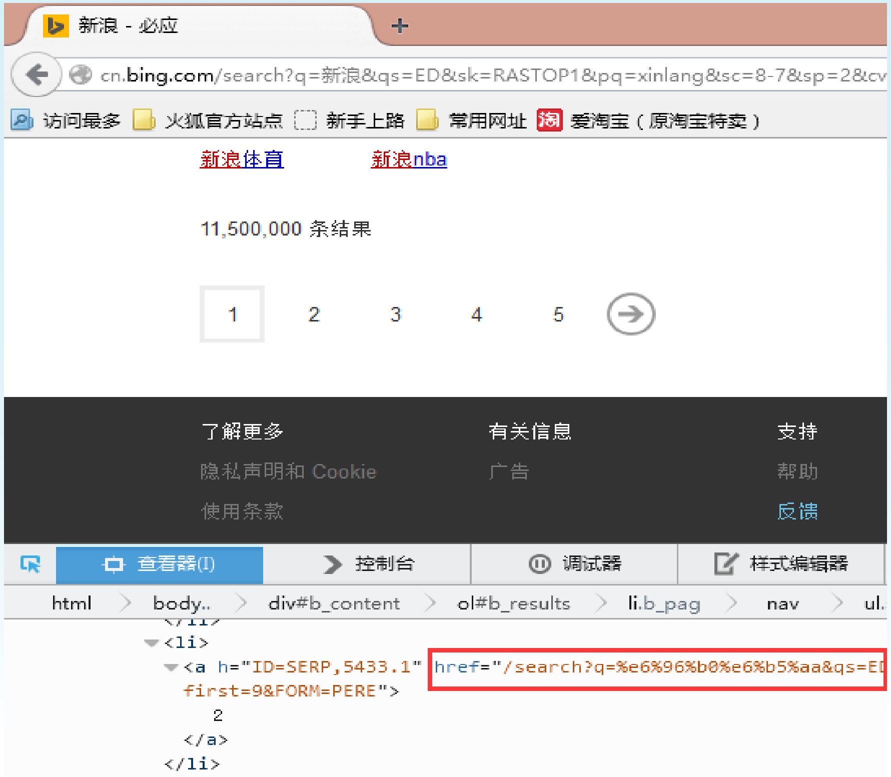
新浪网为全球用户24小时提供全面及时的中文资讯，内容覆盖国内外突发新闻事件、体坛赛事、娱乐时尚、产业资讯、实用信息等，设有新闻、体育、娱乐、财经、科技 ...

www.sina.com.cn ▾ 1 天前

```
<ol id="b_results">
  <li class="b_algo">
    <div class="b_title">
      <div class="b_imagePair square_mi">
        <div class="inner"></div>
        <h2>
          <a h="ID=SERP,5131.2" target="_blank" href="http://www.sina.com.cn/">
            <strong></strong>
             首页
          </a>
        </h2>
```

Write a crawler based on keywords

- How we solve the listing results by page number problem?
  - Regular Expression

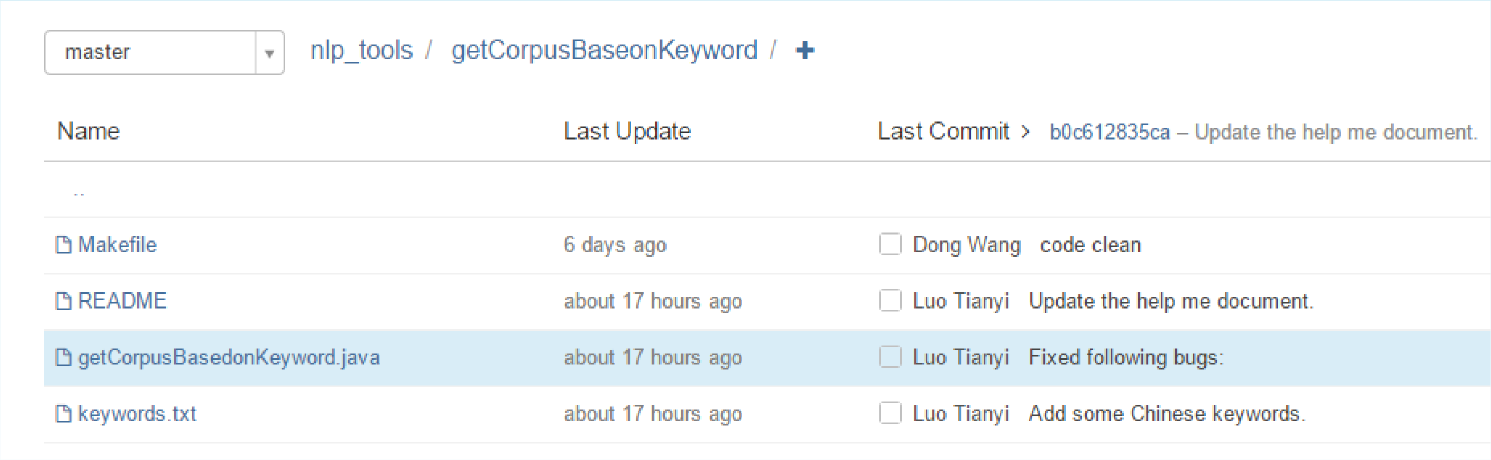


11,500,000 条结果

```
<li>
  <a h="ID=SERP,5433.1" href="/search?q=%e6%96%b0%e6%b5%aa&qs=ED
  first=9&FORM=PERE">
    2
  </a>
</li>
```

Write a crawler based on keywords

- The URL of this crawler based on keywords
  - http://192.168.0.51:8000/speech/nlp_tools/tree/master/getCorpusBaseonKeyword
  - Welcome to use and give me some advice. Thx~



master ▾ nlp_tools / getCorpusBaseonKeyword / +

Name	Last Update	Last Commit › b0c612835ca – Update the help me document.
..		
Makefile	6 days ago	□ Dong Wang code clean
README	about 17 hours ago	□ Luo Tianyi Update the help me document.
getCorpusBasedonKeyword.java	about 17 hours ago	□ Luo Tianyi Fixed following bugs:
keywords.txt	about 17 hours ago	□ Luo Tianyi Add some Chinese keywords.

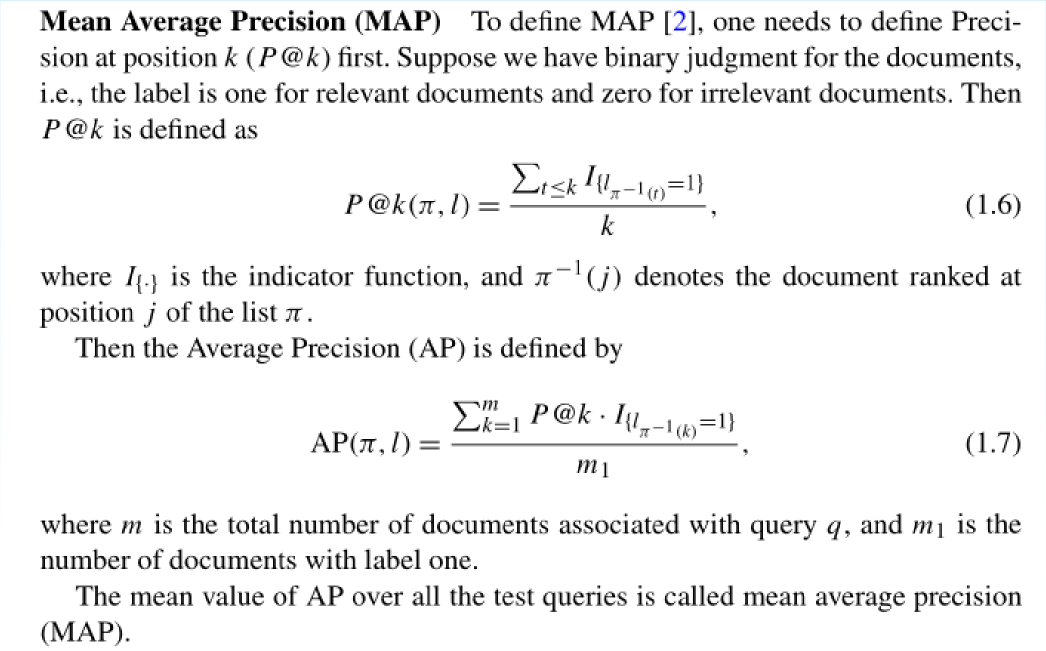
Utilize learning to rank technology to get best QA re-ranking results

- What problem we need to solve?
  - Given candidate sets(50/query) of 1596 queries and their tf * idf score and another score.
  - Utilize learning to rank technology to learn the optimization combination ways of these two scores and conduct re-ranking
  - Data format:

    1 ||| 申请办理高龄老人津贴变更和终止的时限 ||| 86 ||| [{办,办理 }] 高龄 {老人,老年人} 津贴 {变更,终止} [的] {时限,时间限制} [是] [{ 多久,多长,多长时间}] ||| 办理高龄老年人津贴变更和终止的时限？ ||| 户籍迁移及死亡的次月 ||| 3.6066787 2.4930215 ||| 86
  - So this problem is an Information Retrieval problem

Utilize learning to rank technology to get best QA re-ranking results

- Most commonly used evaluation measures of Information Retrieval:
  - Mean Average Precision(MAP) & Precision at position k(P@k)



Mean Average Precision (MAP) To define MAP [2], one needs to define Precision at position k ($P@k$) first. Suppose we have binary judgment for the documents, i.e., the label is one for relevant documents and zero for irrelevant documents. Then $P@k$ is defined as

$$P@k(\pi, l) = \frac{\sum_{t \le k} I_{\{l_{\pi^{-1}(t)}=1\}}}{k}, \tag{1.6}$$

where $I_{\{\cdot\}}$ is the indicator function, and $\pi^{-1}(j)$ denotes the document ranked at position j of the list π.

Then the Average Precision (AP) is defined by

$$\mathrm{AP}(\pi, l) = \frac{\sum_{k=1}^{m} P@k \cdot I_{\{l_{\pi^{-1}(k)}=1\}}}{m_1}, \tag{1.7}$$

where m is the total number of documents associated with query q, and m_1 is the number of documents with label one.

The mean value of AP over all the test queries is called mean average precision (MAP).

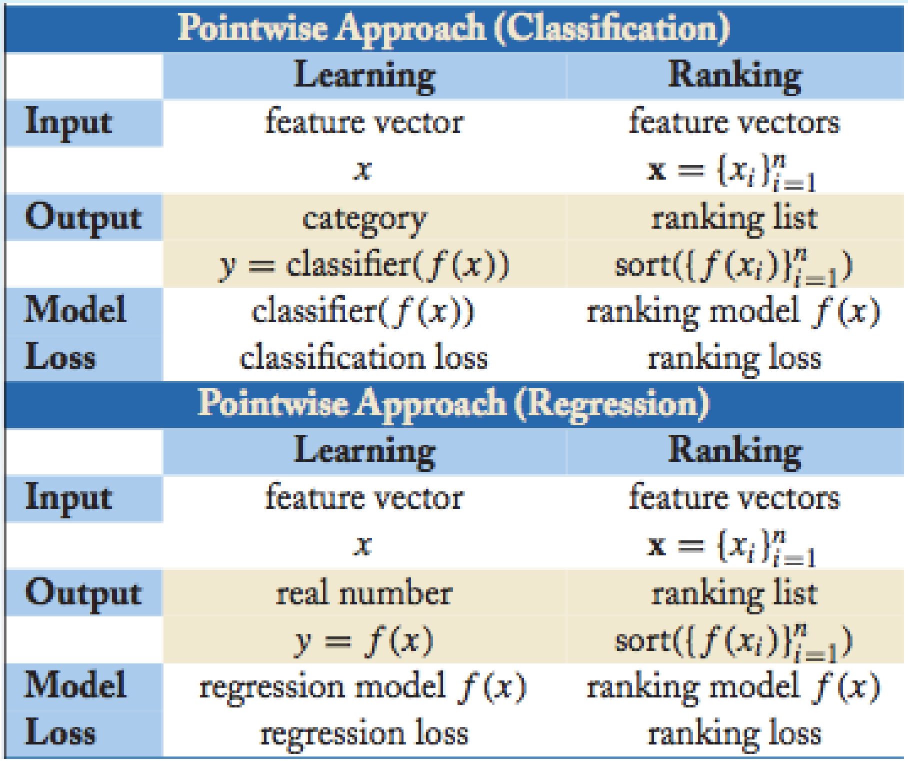
As for our problem, the evaluation measures should be P@1.

Utilize learning to rank technology to get best QA re-ranking results

- Briefly introduction about learning to rank:
  - Two kinds of traditional ranking methods about information retrieval: Relevance Ranking(VSM, TF*IDF and BM25) and Importance Ranking(Pagerank)
  - Disadvantage:
    - Every model only utilize some aspect information about document.
    - If you have many parameters to tune, it is a difficult problem.
  - So we use learning to rank technology to solve ranking problem.

Utilize learning to rank technology to get best QA re-ranking results

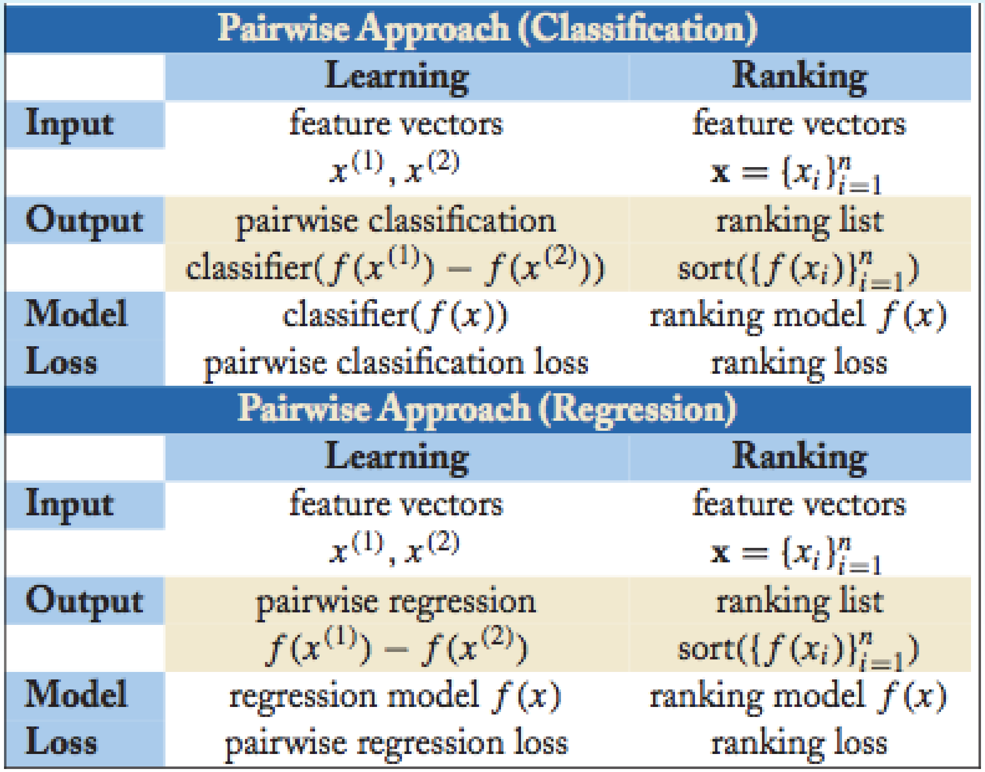
- We group l2r technology into 3 approaches:
  - – 1. Pointwise approach(McRank)



Pointwise Approach (Classification)		
	Learning	**Ranking**
Input	feature vector x	feature vectors $\mathbf{x} = \{x_i\}_{i=1}^{n}$
Output	category $y = \text{classifier}(f(x))$	ranking list $\text{sort}(\{f(x_i)\}_{i=1}^{n})$
Model	$\text{classifier}(f(x))$	ranking model $f(x)$
Loss	classification loss	ranking loss
Pointwise Approach (Regression)		
	Learning	**Ranking**
Input	feature vector x	feature vectors $\mathbf{x} = \{x_i\}_{i=1}^{n}$
Output	real number $y = f(x)$	ranking list $\text{sort}(\{f(x_i)\}_{i=1}^{n})$
Model	regression model $f(x)$	ranking model $f(x)$
Loss	regression loss	ranking loss

Utilize learning to rank technology to get best QA re-ranking results

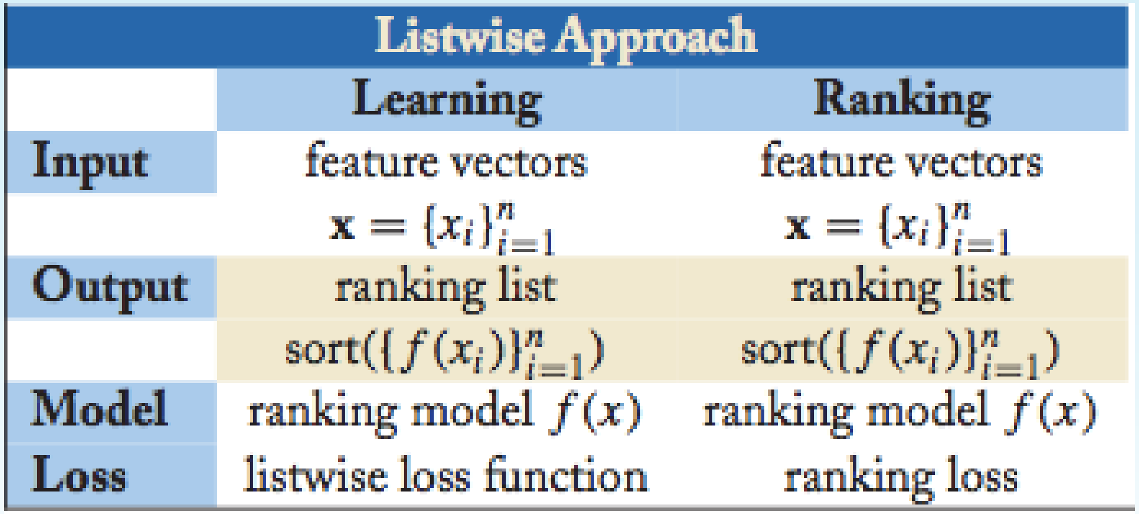
- We group l2r technology into 3 approaches:
  - 2. Pairwise approach()



Pairwise Approach (Classification)		
	Learning	**Ranking**
Input	feature vectors $x^{(1)}, x^{(2)}$	feature vectors $\mathbf{x} = \{x_i\}_{i=1}^{n}$
Output	pairwise classification $\text{classifier}(f(x^{(1)}) - f(x^{(2)}))$	ranking list $\text{sort}(\{f(x_i)\}_{i=1}^{n})$
Model	$\text{classifier}(f(x))$	ranking model $f(x)$
Loss	pairwise classification loss	ranking loss
Pairwise Approach (Regression)		
	Learning	**Ranking**
Input	feature vectors $x^{(1)}, x^{(2)}$	feature vectors $\mathbf{x} = \{x_i\}_{i=1}^{n}$
Output	pairwise regression $f(x^{(1)}) - f(x^{(2)})$	ranking list $\text{sort}(\{f(x_i)\}_{i=1}^{n})$
Model	regression model $f(x)$	ranking model $f(x)$
Loss	pairwise regression loss	ranking loss

Utilize learning to rank technology to get best QA re-ranking results

- We group l2r technology into 3 approaches:
  - 3. Listwise approach



Listwise Approach

	Learning	Ranking
Input	feature vectors $\mathbf{x} = \{x_i\}_{i=1}^n$	feature vectors $\mathbf{x} = \{x_i\}_{i=1}^n$
Output	ranking list $\mathrm{sort}(\{f(x_i)\}_{i=1}^n)$	ranking list $\mathrm{sort}(\{f(x_i)\}_{i=1}^n)$
Model	ranking model $f(x)$	ranking model $f(x)$
Loss	listwise loss function	ranking loss

Utilize learning to rank technology to get best QA re-ranking results

- The results we get utilizing RankLib(A open sourced toolkit):

– MART:	0.6867167919799498
– RankNet:	0.6911027568922306
– RankBoost:	0.6867167919799498
– AdaRank:	0.6873433583959899
– LambdaMART:	0.6735588972431078
– ListNet:	0.6704260651629073

Want to do next week

- Define more features to add l2r framework to train a model which could provide higher accuracy rate about QA system.
- Continuously update the crawler based on keywords.

Thank You !