

Soccer Information Search Engine with Direct-reply Component

Dong Keyang: 3130000020, 517908768@qq.com

Wang Ru: 3120102084, rockidog@live.com

Fang Qixiang: 3120101860, rafefang@163.com

ABSTRACT

The contribution of our information retrieval project is a general search engine which is specialized in soccer news, and is equipped with a question-like query disposing module. Besides the implementing of classic algorithms in information retrieval such as tf-idf weight ranking and inverted index, we also realized a basic version of natural language parsing algorithm so that the extra tool is enabled to guess the exact answer wanted by any primary user. The crawler and server-end are established over existing widely-used frameworks in Python, named Scrapy and Django.

1 INTRODUCTION

The need for grepping informative or instructive search results more efficiently is always crucial for primary Internet users. Users may ask the server end for an information retrieval engine or tool natural language questions as if they were talking to an human expert; thus, in matching to the realistic requirements, our search engine should recognize various patterns of question-like queries, and guess the most probable answer the user may want by applying customized linguistic parsing algorithms for the very pattern. Famous Q&A websites such as Quora, Stackoverflow decides the ranking of answers by users' voting, which is of high precision, but lack intelligence and do not match our aim well. Therefore we turned to develop an extra component in a general search engine, which is supported by a database specialized in soccer news worldwide.

2 A DETAILED REVIEWING

2.1 Main Function

As we proposed earlier, the periodic achievement of our group's working on project turns out to be a general information retrieval engine with enhanced components. Besides providing with reasonably ranked search results, which include URL, title and a snapshot of the original webpage in every item, it can dispose some patterns of question-like query(complete or incomplete sentences in pure English only), and thus can also provide an exact answer through algorithmic guessing for users.

The major advantage of the product(as well as its essential idea) is obvious: those non-professional users who tend to send natural language sentences as queries to search engines will also get the wanted information efficiently; and the attempt to add the direct-reply new characteristic will also benefit those people who need bunches of instructive answers in very limited time. In other word, it could be another step towards developing a more user-friendly search engine/tool. Due to the limitation of sample data's range, the asking and answers should be constrained to soccer news, even instructive answers are trimmed.

2.2 Review of Past Analogous Products and Researches

Mainstream search engines such as Google or Baidu have already developed similar direct-informing function in recent years: try typing high-frequency keywords like “weather” in the search bar, Baidu will not only give prediction and probable completion on your typed query(Figure 1), but also shows a specialized block consists of useful, detailed information relating to “weather”(Figure 2).



Figure 1



Figure 2

Although Baidu is trying to guess the true requirement of searchers to some extent, it is still some distance from the need of primary users on Internet; for example, when we alter the query to “weather”(Figure 3), the guessing mechanism does not work anymore. You may notice that Baidu usually gives terms found in Zhidao, Baidu Music, News or anything else they build by themselves rather than collected a higher rank; And that is not part of the new component we are discussing here. Therefore, noticing the lack of natural language supporting, our project made efforts to improve

the search engine's performance in the very aspect.



Figure 3

2.3 Making Use of Existing Resources

2.3.1 Data Sources

In order to form more compact index and more specified, easy recognized query results, we majored our data sources in the field of football game news, tested those data crawled from Yahoo! Soccer as well as ESPN, and finally focused on one single topic, which is FIFA World Cup 2014. Our crawler is customized for the site FIFA.com, from which the snatched date is quite enough for various testing works.

2.3.2 Technical Basis

Using the mature and efficient environment for developers, Python 2.7, we made use of structures of two open-source projects on Github: Scrapy-0.25 for constructing the crawler, and Django-1.6.5 for simulating a server as the user interface component and the major dynamic part.

Scrapy v0.25

The most significant technique we used for scratching data is Scrapy, a fast high-level framework for web scraping and structural data extracting based on Python. It can be used for a wide range of purposes, from data mining to monitoring and automated testing. Comparing to original libraries "urllib" and "urllib2", Scrapy can offer a lighter way to fulfill the same job.

Why scrapy? We have considered the python libraries of liburl or liburl2. By comparison, scrapy offered a more easy and effective way to do the same job.

We main crawl for some specific content from the web pages. Here is a quick view at our expected content:

```
import scrapy
class DmozItem(scrapy.Item):
    title=scrapy.Field()
    link=scrapy.Field()
    desc=scrapy.Field()
```

As is shown above, DmozItem is the basic item class inherited from scrapy.Item for scratching a web

page. It will be used to store the title, link as well as description of a page.

Figure 4 indicates the logical diagram of Scrapy.

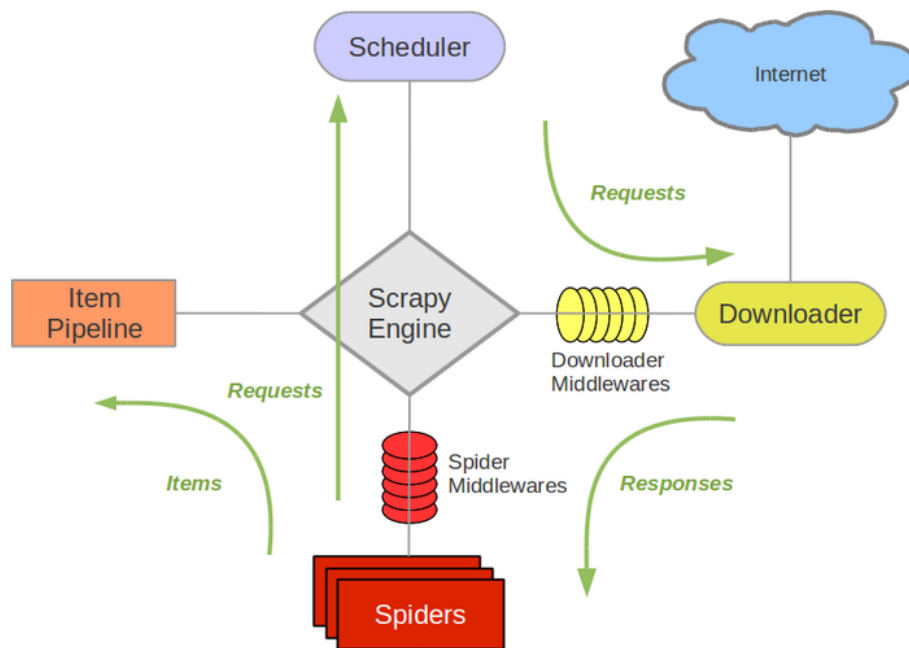


Figure 4

For more information on scrapy see the appendix or read the [documentation](#). online.

Django v1.6.5

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It is designed to handle two challenges: the intensive deadlines of a newsroom and the stringent requirements of the experienced Web developers who wrote it.

That suggests the reason why we apply the very technique in the project to simulate a server environment free of any real web server software or database support.

The layout of a Django-1.6.5 project is demonstrated as following, Figure 5 and Figure 6. A Simple command can produce the framework for developers, and the essential parts are settings.py, urls.py and views.py(created myself as a library of functions). The main philosophy of Django the framework emphasizes the separation of workloads; to be specific, urls.py decides which function in views.py should be called to produce the content of page in corresponding to certain URL accesses, which is easy to handle. For primary works it is no need to add too much to setting.py, except that we must tell where the Web templates are(in this project they are placed simply under the working directory "mysite", as search_form.html does).

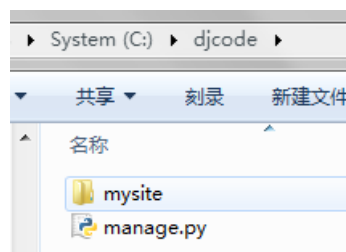


Figure 5

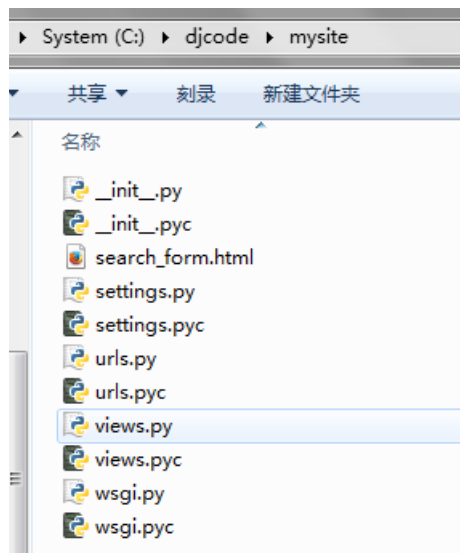


Figure 6

2.4 Essential Algorithms or Data Structures

2.4.1 Ranking

We adopt the classic tf-idf weighting algorithm to decreasing rank the search result; the formula is showed as the following:

where the value is specifically for the tuple(term, document), and W equals to zero when $tf=0$; the condition that $df=0$ will not occur, because a term that appears in no document will not be indexed.

In a word, the weight of certain document in correspond to a query increases with the number of occurrences within a document(term frequency), and also increases with the rarity of the term in the collection(inverse document frequency).

2.4.2 Indexing Format

Each term appeared in disposed documents that produced by web crawler will occupy a single file as its indexing file, taking the term itself as its file name; inside the index file, each line has identical format as:

```
Document_number : term_frequency : first_line_number : ... : last_line_number
```

When a certain term is referred in user's query request, the rear-end which accepted it through "GET" method from the fore-end of web server will manage to locate the very index file in a certain directory, reads the indexing information it stored into the process' memory and suits the data structure below:

```
{term1:[[doc1,tf,line1,line2,...],[doc2,...]],term2:...}
```

which is a nesting of built-in dictionary and list in Python programming language, and could be very easy and clear for further analyzing.

2.4.3 Disposal of Natural Language Query

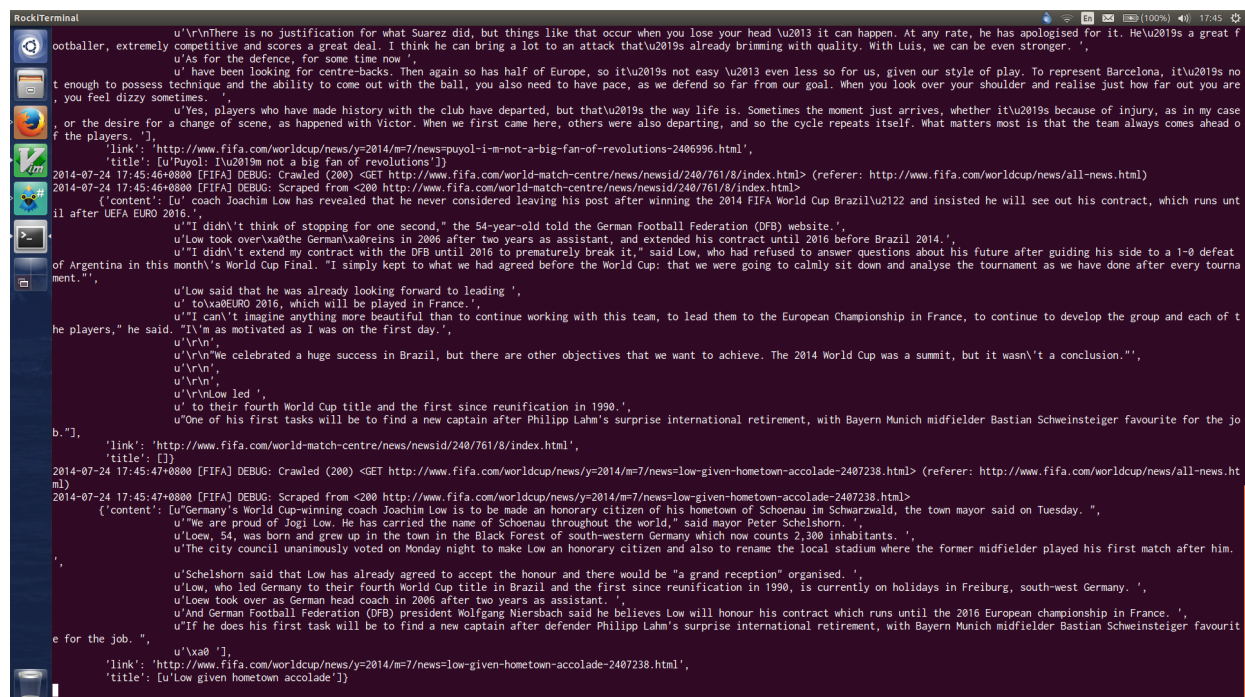
As handling natural language query input is a quite complex topic, we merely made a few steps towards the ultimate aim. The query filter applies a chart of stop words that are identical to those used for building inverted-index, and record the English question words such as "when", "where", "why", "what", "which", "how". To match higher standard, phrases like "how much" may also be included, but we don't think any search engine can judge users' yes-or-no questions like "is...?" "could there be...?".

The sentence is converted into keywords plus a question pattern through the technique that addressed above; after the keywords are indexed, intersected and then ranked to provide normal search results, the question pattern we got decides which algorithm will be applied to parse the top ranked documents for a second time to find the probable most wanted answer. We left the interface of question patterns, and implemented one simple algorithm for evaluating: pick a document of high rank, then try selecting a reliable instance of keyword and try to get a phrase that is within in k-words distance to it in the same sub-sentence. In our alpha version, the pattern of phrase is proper nouns that consists of words starting with upper-case letters, which could be matched using regular expressions.

2.5 Overview of the Overall Search Engine

2.5.1 Simple Tutorial

Web pages crawling, index building should be done in sequence before any query is available for testing. Simply run the Python script crawler.py placed at any path by interpreters like IDLE will do, and snatching pages can take long time, as shown in Figure 7.



```

RockTerminal
u'\nThere is no justification for what Suarez did, but things like that occur when you lose your head \u2013 it can happen. At any rate, he has apologised for it. He\u2019s a great footballer, extremely competitive and scores a great deal. I think he can bring a lot to an attack that\u2019s already brimming with quality. With Luis, we can be even stronger. ',
u'As for the defence, for some time now ',
u' have been looking for centre-backs. Then again so has half of Europe, so it\u2019s not easy \u2013 even less so for us, given our style of play. To represent Barcelona, it\u2019s not enough to possess technique and the ability to come out with the ball, you also need to have pace, as we defend so far from our goal. When you look over your shoulder and realise just how far out you are, you feel dizzy sometimes. ',
u'Yes, players who have made history with the club have departed, but that\u2019s the way life is. Sometimes the moment just arrives, whether it\u2019s because of injury, as in my case or the desire for a change of scene, as happened with Victor. When we first came here, others were also departing, and so the cycle repeats itself. What matters most is that the team always comes ahead of the players. ',
'link': 'http://www.fifa.com/worldcup/news/y=2014/m=7/news=puyol-i-m-not-a-big-fan-of-revolutions-2406996.html',
'title': '[u]Puyol: \u2018I\u2019m not a big fan of revolutions\u2019']
2014-07-24 17:45:46+0800 [FIFA] DEBUG: Crawled (200) <GET http://www.fifa.com/world-match-centre/news/newsid/240761/8/index.html> (referer: http://www.fifa.com/worldcup/news/all-news.html)
2014-07-24 17:45:46+0800 [FIFA] DEBUG: Scraped from <200 http://www.fifa.com/world-match-centre/news/newsid/240761/8/index.html>
{'content': '[u] coach Joachim Low has revealed that he never considered leaving his post after winning the 2014 FIFA World Cup Brazil\u2013 and insisted he will see out his contract, which runs until after UEFA EURO 2016. ',
u' "I didn\u2019t think of stopping for one second," the 54-year-old told the German Football Federation (DFB) website. ',
u'Low took over \u2013 the German \u2013 in 2006 after two years as assistant, and extended his contract until 2016 before Brazil 2014. ',
u' "I didn\u2019t extend my contract with the DFB until 2016 to prematurely break it," said Low, who had refused to answer questions about his future after guiding his side to a 1-0 defeat of Argentina in this month\u2019s World Cup Final. "I simply kept to what we had agreed before the World Cup: that we were going to calmly sit down and analyse the tournament as we have done after every tournament." ',
u'Low said that he was already looking forward to leading ',
u' "I can\u2019t imagine anything more beautiful than to continue working with this team, to lead them to the European Championship in France, to continue to develop the group and each of the players," he said. "I\u2019m as motivated as I was on the first day. ',
u'\n',
u'\nWe celebrated a huge success in Brazil, but there are other objectives that we want to achieve. The 2014 World Cup was a summit, but it wasn\u2019t a conclusion." ',
u'\n',
u'\n',
u'\nLow led ',
u' to their fourth World Cup title and the first since reunification in 1990. ',
u'One of his first tasks will be to find a new captain after Philipp Lahm\u2019s surprise international retirement, with Bayern Munich midfielder Bastian Schweinsteiger favourite for the job. ',
'link': 'http://www.fifa.com/world-match-centre/news/newsid/240761/8/index.html',
'title': '[ ]'}
2014-07-24 17:45:47+0800 [FIFA] DEBUG: Crawled (200) <GET http://www.fifa.com/worldcup/news/y=2014/m=7/news=low-given-hometown-accolade-2407238.html> (referer: http://www.fifa.com/worldcup/news/all-news.html)
2014-07-24 17:45:47+0800 [FIFA] DEBUG: Scraped from <200 http://www.fifa.com/worldcup/news/y=2014/m=7/news=low-given-hometown-accolade-2407238.html>
{'content': '[u]Germany\u2019s World Cup-winning coach Joachim Low is to be made an honorary citizen of his hometown of Schoenau im Schwarzwald, the town mayor said on Tuesday. ',
u' "We are proud of Jogi Low. He has carried the name of Schoenau throughout the world," said mayor Peter Schelshorn. ',
u'Low, 54, was born and grew up in the town in the Black Forest of south-western Germany which now counts 2,300 inhabitants. ',
u'The city council unanimously voted on Monday night to make Low an honorary citizen and also to rename the local stadium where the former midfielder played his first match after him. ',
u'Schelshorn said that Low has already agreed to accept the honour and there would be "a grand reception" organised. ',
u'Low, who led Germany to their fourth World Cup title in Brazil and the first since reunification in 1990, is currently on holidays in Freiburg, south-west Germany. ',
u'Low took over as German head coach in 2006 after two years as assistant. ',
u'And German Football Federation (DFB) president Wolfgang Niersbach said he believes Low will honour his contract which runs until the 2016 European championship in France. ',
u'If he does his first task will be to find a new captain after defender Philipp Lahm\u2019s surprise international retirement, with Bayern Munich midfielder Bastian Schweinsteiger favourite for the job. ',
u'\xa0 ',
'link': 'http://www.fifa.com/worldcup/news/y=2014/m=7/news=low-given-hometown-accolade-2407238.html',
'title': '[u]Low given hometown accolade']

```

Figure 7

As for the indexer, it is better to place it under the working directory of our Django project `C:/djcode/mysite` on my Windows 7, which includes `/doc` containing the pages crawled. Running the script will create thousands of index files under `/index`. Figure 8 gives a snapshot while processing the analysis and indexing period.



Figure 8

To start sending queries for testing, we need to get into Command Line (instruction “cmd” in Windows), type in `python [working-dir]/manage.py runserver [port-num]`. The default value of port number is 8000, and you may ignore the setting of it. See Figure 9, Django is now available.

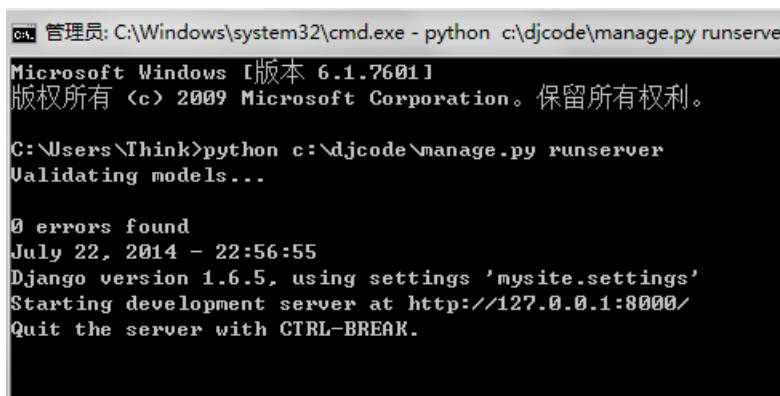


Figure 9

Start the web browser and get to `http://localhost:8000/search-form`, searching can now begin; result page will be at `/search`, which will automatically be linked.

2.5.2 Logical Structure of Components

As is shown in Figure 10, the project can be divided into several components that work separately as an orgasm: a Web Crawler based on Scrapy, an Inverted-Index builder and a server component based on Django that shoulders dynamic tasks in respond to users’ actions.

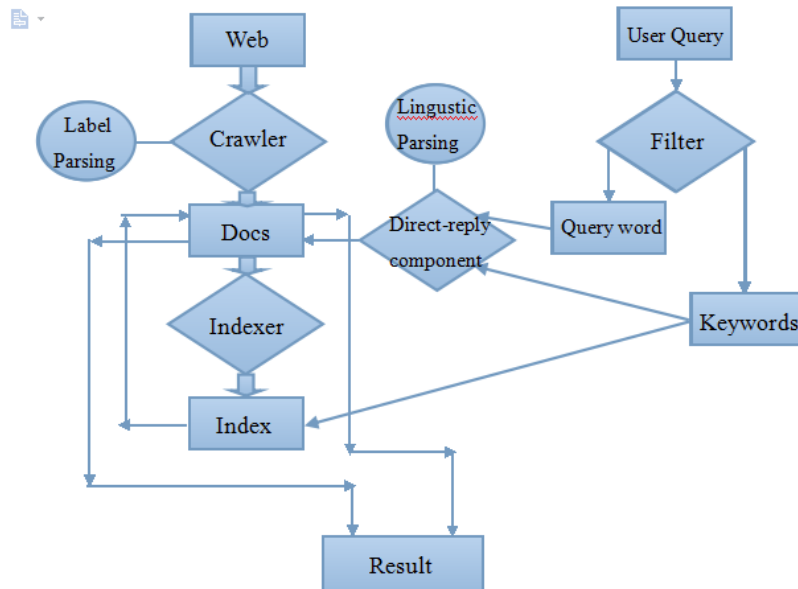


Figure 10

2.6 Challenges and Details

It is hard to decide which punctuation should be deleted. For example, if we delete the “-” in compound words like “well-known”, they will split into discrete words. Recognize the marks that split word from those link between words are quite sophisticated.

The encoding rules are different between python and documents downloaded by crawler. For example, the hard space’s code is `/0xC2 /0xA0` in UTF-8; even python can read it, we can’t make any use of it in python, but only causes mysterious characters in index names. Finally, we decided to use ASCII, and meanwhile applying a routine to change any char-type with a code larger than 127 into a space.

Upper-case letters and lower-case letters are of no difference in Windows file names, so we store only lower-case versions of all indexed terms.

The main challenge we met while crawling web pages is the strategy to determine which urls to follow. At most time is unexpected to follow non-relevant advertisement urls. Our solution is to limit the url domain and apply different ways on different types of pages. Actually it is a matter of time to do the job of accurate web-craping.

2.7 Performance Evaluation

At first our aims include establishing the user interface like a naive text message version SIRI, which appeared to be a famous vocal AI in IOS mobile systems, and can always provide decent reply on users’ questions or guess users’ preference on searching even if the database is not well-formed enough. But to this stage actually, our tool may only provide with a simple informative answer in extra, which cannot be accurate in many occasions due to the lack of optimization in algorithm.

Due to the choosing of question-like query can by no means be random enough, and the original documents are from a specific field, it remained a tough problem to evaluate to what extent does our direct-reply algorithm make sense. Through limited testing and optimizing, we found that the algorithm tends to provide acceptable answers when it is short (less than two words) , and dramatically loses it’s precision as the length of proper noun answer increases.

3 CONCLUSIONS

It can be foreseen that constructing and maintaining a pattern library of high-quality should greatly increase the accuracy of direct-reply answers. Further development of this library shall not only include regular expression matching technique, but also apply linguistic parsing mechanisms that suits various query patterns(asking for time, location, even reasons, etc).

Furthermore, wildcard searching or spelling correction is not implemented yet. To realize the so called "vague searching", a B-tree indexing structure is required; but spelling correction is easy to be added as an extra and necessary feature, simply by using a well-performed stemmer together with collected term frequency data.

4 REFERENCES

<http://www.djangobook.com/en/2.0/index.html>
<http://doc.scrapy.org/en/latest/>
<http://sports.yahoo.com/soccer/>
<http://www.espnfc.com>
<http://www.fifa.com>
<http://nlp.stanford.edu/IR-book/html/htmledition/irbook.html>

5 APPENDIX

Here provides the foundation of scrapy.

```
import scrapy
from tutorial.items import DmozItem

class DmozSpider(scrapy.Spider):
    name = "dmoz"
    allowed_domains = ["dmoz.org"]
    start_urls = [
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Books/",
        "http://www.dmoz.org/Computers/Programming/Languages/Python/Resources/"
    ]

    def parse(self, response):
        for sel in response.xpath('//ul/li'):
            item = DmozItem()
            item['title'] = sel.xpath('a/text()').extract()
            item['link'] = sel.xpath('a/@href').extract()
            item['desc'] = sel.xpath('text()').extract()
            yield item
```

The class `DmozSpider` defined the spider we used for scratching. Starting from the two urls, it will returned the item instance of every page it pass through. The item returned will be processed by a pipeline mechanism provided by scrapy. To print out or to store the information is up to you.