

# DBWorld 搜索引擎

吴紫薇 PB16110763

## 实验内容

为DBWorld (<https://research.cs.wisc.edu/dbworld/browse.html>) 设计并开发一个搜索引擎。

## 实验环境

Linux Ubuntu 16.04

Python + Flask web framework

Libs: selenium, whoosh, nltk

## 实验步骤及方法

1. 抓取DBWorld信息 (mycrawler.py class DBworldCrawler)

使用selenium加载firefox webdriver,初始化类, 根据日期时间新建一个文件夹以保存爬取的文件。

```
def __init__(self, menu_url="https://research.cs.wisc.edu/dbworld/browse.html",
output_dir="messages"):
    self.menu_url = menu_url
    self.output_dir = output_dir
    # load browser driver
    opt = webdriver.FirefoxOptions()
    # no display mode
    opt.add_argument('--headless')
    self.browser = webdriver.Firefox(firefox_options=opt)
    if DEBUG_INIT: print "Driver loaded."

    # make a new dir to store jsons
    self.new_dir_path = "{}/{}/".format(self.output_dir,
datetime.now().strftime("%Y%m%d%H%M%S"))
    os.mkdir(self.new_dir_path)
```

抓取全部消息, 通过分析网页代码发现消息由“TBDY”这一tag标识, 每个信息由“TD”标识, 从而抓取到包括sent日期, type类型, author作者, subject主题, href详情链接, deadline截止日期, webpage网页链接; 我们把这些信息以json文件的形式保存下来。

```
def crawl_menu_url(self):
    self.browser.get(self.menu_url)
    msgs = self.browser.find_elements_by_tag_name("TBDY")
    # process all messages, table contents
    p = progressbar.ProgressBar()
    p.start()
    print "Crawling messages from {}".format(self.menu_url)
    total_num = len(msgs)
```

```

        for idx, msg in enumerate(msgs):
            TDs = msg.find_elements_by_tag_name("TD")
            Sent = TDs[0].text
            Type = TDs[1].text
            Author = TDs[2].text
            Subject = TDs[3].text
            tmp = TDs[3].find_element_by_tag_name("A")
            Detail_href = tmp.get_attribute("HREF")
            Deadline = TDs[4].text
            try:
                WebPage_href =
TDs[5].find_element_by_tag_name("A").get_attribute("HREF")
            except Exception as e:
                WebPage_href = ""
            if DEBUG_MENU : print "
{}\\t{}\\t{}\\t{}\\t{}\\t{}\\n".format(Sent,Type,Author,Subject,Detail_href,Deadline,
WebPage_href)

            # save as json
            out_path = "{}/{}/{}.json".format(self.new_dir_path, idx)
            out_dict = {
                "sent": Sent,
                "type": Type,
                "author": Author,
                "subject": Subject,
                "href": Detail_href,
                "deadline": Deadline,
                "webpage": WebPage_href
            }

            with open(out_path,"w") as out_file:
                out_file.write(json.dumps(out_dict))

            p.update(idx*100 / total_num)

p.finish()

```

完成消息爬取后，进行详情信息爬取；先从之前保存好的json文件中读出href信息，再爬取该网页内容，通过分析网页代码发现，content内容由“PRE”这一tag标识；将详情内容保存成txt文件。

```

def crawl_detail_url(self):
    jsongdir=self.new_dir_path
    total_num = len(os.listdir(jsongdir))
    p = progressbar.ProgressBar()
    p.start()
    print "Crawling detail contents of messages."
    for idx in range(total_num):
        # load json
        jsonpath = "{}/{}/{}.json".format(jsongdir, idx)
        with open(jsonpath, "r") as in_file:
            buf = in_file.read()
            dic_data = json.loads(buf)
            url = dic_data["href"]

```

```

        try:
            self.browser.get(url)
            content = self.browser.find_element_by_tag_name("PRE").text
        except Exception:
            content = ""

        with open("{} / {}.txt".format(self.new_dir_path, idx), "w") as out_file:
            out_file.write(content.encode("utf-8"))

        p.update(idx*100 / total_num)
    p.finish()

```

## 2. 建立索引文件及检索程序 (myindexer.py class DEworldIndexer)

首先建立索引文件的schema

```

schema = Schema(
    author=TEXT(stored=True),
    sent=DATETIME(stored=True, sortable=True),
    deadline=DATETIME(stored=True, sortable=True),
    subject=TEXT(stored=True),
    content=TEXT(stored=True),
    doctype=TEXT(stored=True),
    href=TEXT(stored=True),
    webpage=TEXT(stored=True)
)

```

读取保存好的json文件和txt文件，抽取出包括时间、地点、会议名称等关键信息，构建索引。

```

for idx in range(self.docnum):
    jsonpath = "{} / {}.json".format(self.msgdir, idx)
    txtpath = "{} / {}.txt".format(self.msgdir, idx)

    in_json = open(jsonpath, "r")
    buf = in_json.read()
    dic_data = json.loads(buf)
    #print(dic_data)

    in_txt = open(txtpath, "r")
    txt_data = in_txt.read()
    #print(txt_data)

    sent_tmp = dic_data["sent"]
    if len(sent_tmp):
        sent_tmp = sent_tmp.split("-")
        sent_field = "{}-{}-{}".format(sent_tmp[2], mon2num[sent_tmp[1]],
sent_tmp[0])

    deadline_tmp = dic_data["deadline"]
    if len(deadline_tmp):
        deadline_tmp = deadline_tmp.split("-")

```

```

        deadline_field = "{}-{}-{}".format(deadline_tmp[2],
mon2num[deadline_tmp[1]], deadline_tmp[0])
    else:
        deadline_field = None

    #print(sent_field)
    #print(deadline_field)
    if deadline_field:
        writer.add_document(
            author=dic_data["author"],
            sent=sent_field,
            deadline=deadline_field,
            subject=dic_data["subject"],
            content=txt_data,
            doctype=dic_data["type"],
            href=dic_data["href"],
            webpage=dic_data["webpage"]
        )
    else:
        writer.add_document(
            author=dic_data["author"],
            sent=sent_field,
            subject=dic_data["subject"],
            content=txt_data,
            doctype=dic_data["type"],
            href=dic_data["href"],
            webpage=dic_data["webpage"]
        )

    print("{} added".format(txtpath))
    # commit adding process
    writer.commit()

```

基于whoosh的检索程序，根据前端选择的不同检索域对不同的信息进行检索。

```

class DBworldSearcher:

    def __init__(self, indexdir, fieldlist=["subject", "content"]):
        self.indexdir = indexdir
        ix = open_dir(indexdir)

        #self.parser = QueryParser("subject", self.ix.schema)
        self.parser = MultifieldParser(fieldlist, ix.schema)
        self.parser.add_plugin(DateParserPlugin())
        self.searcher = ix.searcher()

    def search(self, querytext, limit):
        myquery = self.parser.parse(querytext)
        results = self.searcher.search(myquery, limit=limit)
        return results

```

### 3. 服务器和网页开发

主要包括主页 (mainpage.html) 和搜索详情页 (results.html)

通过一个form传递query文本和目标检索域, 以request args的形式传递给search函数。

```
@app.route('/', methods=["GET", "POST"])
def mainpage():
    # POST and query not empty
    if request.method == "POST" and len(request.form["query"]):
        query = request.form["query"]
        filedid = request.form["field"]
        #print(filedid)
        return redirect(url_for('search', q=query, p=1, f=filedid))
    # GET
    return render_template("mainpage.html")

@app.route('/search', methods=["GET", "POST"])
def search():
    # POST a new query
    if request.method == "POST":
        query = request.form["query"]
        filedid = request.form["field"]
        return redirect(url_for('search', q=query, p=1, f=filedid))

    # Search query
    query = request.args["q"]
    page = int(request.args["p"])
    filedid = request.args["f"]

    if filedid == "0":
        # search subject & content
        dbworld_searcher = sub_con_searcher
        tmp = dbworld_searcher.search(querytext=query, limit=page*10)
        time_cost = round(tmp.runtime, 3)
        results = [(x["sent"], x["author"], x["subject"],
                    x["deadline"], x.highlights("content"), x["href"], x["webpage"],
                    x["doctype"]) for x in tmp]
        return render_template("results.html",
                               msg=[len(tmp), time_cost], query=query, page=page, results=results)

    elif filedid == "1":
        # search author
        dbworld_searcher = auth_searcher
        tmp = dbworld_searcher.search(querytext=query, limit=page*10)
        time_cost = round(tmp.runtime, 3)
        results = [(x["sent"], x["author"], x["subject"],
                    x["deadline"], x["content"][:600], x["href"], x["webpage"],
                    x["doctype"]) for x in tmp]
        return render_template("results.html",
                               msg=[len(tmp), time_cost], query=query, page=page, results=results)

    elif filedid == "2":
        # search conference
        dbworld_searcher = conf_searcher
        tmp = dbworld_searcher.search(querytext=query, limit=page*10)
```

```

        time_cost = round(tmp.runtime, 3)
        results = [(x["sent"], x["author"], x.highlights("subject"),
                    x["deadline"], x["content"][:600], x["href"], x["webpage"],
                    x["doctype"])) for x in tmp]
        return render_template("results.html",
                               msg=[len(tmp), time_cost], query=query, page=page, results=results)

    elif filedid == "3":
        # search sent date
        dbworld_searcher = sent_searcher
        tmp = dbworld_searcher.search(querytext=query, limit=page*10)
        time_cost = round(tmp.runtime, 3)
        results = [(x["sent"], x["author"], x["subject"],
                    x["deadline"], x["content"][:600], x["href"], x["webpage"],
                    x["doctype"])) for x in tmp]
        return render_template("results.html",
                               msg=[len(tmp), time_cost], query=query, page=page, results=results)

    elif filedid == "4":
        # search ddl date
        dbworld_searcher = ddl_searcher
        tmp = dbworld_searcher.search(querytext=query, limit=page*10)
        time_cost = round(tmp.runtime, 3)
        results = [(x["sent"], x["author"], x["subject"],
                    x["deadline"], x["content"][:600], x["href"], x["webpage"],
                    x["doctype"])) for x in tmp]
        return render_template("results.html",
                               msg=[len(tmp), time_cost], query=query, page=page, results=results)

```

## 实验结果说明及演示

爬虫运行结果

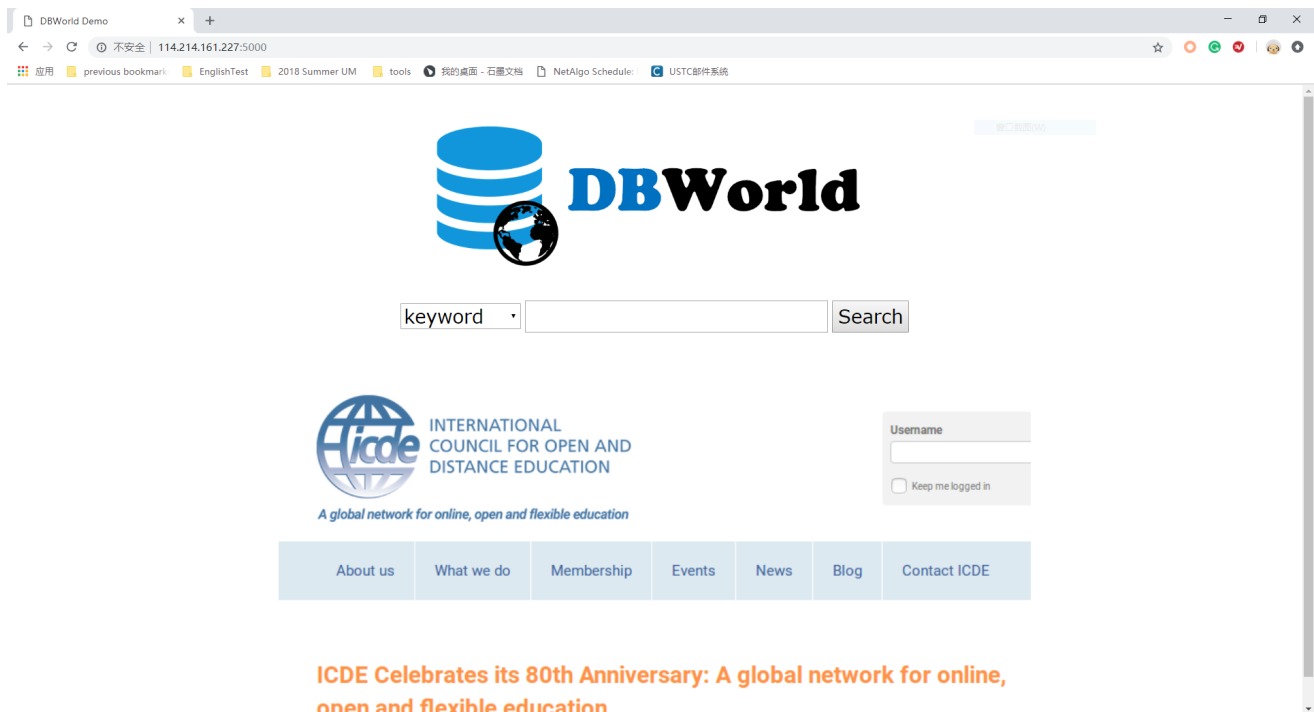
```
yuanmu@Mubuntu: ~/workspace/DBworld_search_engine
DBworld_search_engine % ipython mycrawler.py
Driver loaded.
Crawling messages from https://research.cs.wisc.edu/dbworld/browse.html
100% |#####|
Finished in 92.0353038311 s
Crawling detail contents of messages.
100% |#####|
Finished in 8202.10101914 s
DBworld_search_engine %
```

web服务器运行及处理请求结果

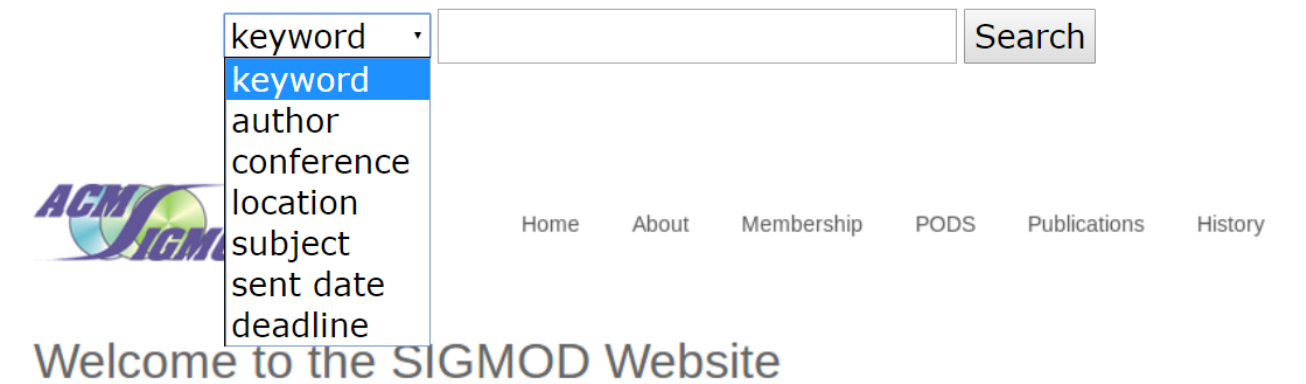
```
flask run -h 0.0.0.0 -p 5000
(venv) DBworld_search_engine % flask run -h 0.0.0.0 -p 5000
* Serving Flask app "demo.py" (lazy loading)
* Environment: development
* Debug mode: on
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 101-230-523
127.0.0.1 - - [14/Dec/2018 15:32:51] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [14/Dec/2018 15:32:51] "GET /favicon.ico HTTP/1.1" 404 -
222.195.92.121 - - [14/Dec/2018 15:32:59] "GET /search?p=5&f=0&q=database HTTP/1.1" 200 -
```

demo网页 <http://114.214.161.227:5000/> 使用校园网访问

主页:



多种检索域选择:



检索会议&期刊名 conference: VLDB



DBWorld Demo x +

114.214.161.227:5000/search?f=2&q=VLDB&p=1

应用 previous bookmark EnglishTest 2018 Summer UM tools 我的桌面 - 石墨文档 NetAlgo Scheduler USTC邮件系统

DBWorld conference VLDB search

5 results retrieved in 0.001s.

[conf. ann.] **VLDB** 2019 : Call For Tutorial Proposals

2018-12-02 [release] ---- 2019-03-12 [ddl] By Sumita Barahmand Barahmand

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-12/1543786028.html>

Webpage : <http://vldb.org/2019/?call-for-tutorials>

----- Call For Tutorials -----

----- 45th International Conference on Very Large Data Bases (VLDB 2019) <http://vldb.org/2019/?call-for-tutorials> -----

[conf. ann.] **VLDB** 2019 : Call For Demonstration Proposals

2018-09-17 [release] ---- 2019-03-15 [ddl] By Sumita Barahmand Barahmand

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-09/1537161993.html>

Webpage : <http://vldb.org/2019/?call-for-demonstrations>

----- Call For Contributions - Demonstrations -----

----- 45th International Conference on Very Large Data Bases (VLDB 2019) <http://vldb.org/2019/?call-for-demonstrations> Los A -----

[journal ann.] The **VLDB** Journal || Vol. 27 issue

2018-11-28 [release] ---- TBD [ddl] By Annette Hinze

检索作者author: alex

DBWorld Demo x +

114.214.161.227:5000/search?p=1&q=alex&f=1

应用 previous bookmark EnglishTest 2018 Summer UM tools 我的桌面 - 石墨文档 NetAlgo Scheduler USTC邮件系统

DBWorld author alex search

1 results retrieved in 0.0s.

[job ann.] Researcher – Software

2018-10-07 [release] ---- 2018-10-30 [ddl] By Alex Vakaloudis

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-10/1538918625.html>

Webpage : <http://www.nimbus.cit.ie/>

2 Year Fixed-Term. Whole-Time or Part-Time. Cork Institute of Technology invites applications from suitably qualified persons for the position of Researcher at the Nimbus Research Centre. About Nimbus: With a staff of over 70, the Nimbus Research Centre, CIT, is Ireland's leading research centre in the Internet of Things (IoT) and Cyber Physical Systems domain. From energy and water optimisation and conservation to smart city infrastructure and Industry 4.0, the Nimbus Centre, in collaboration with partners from all corners of the globe, plays a key role in harnessing and translating the re

<< 1 >>

检索会议地点 location: paris france

DBWorld Demo

114.214.161.227:5000/search?p=1&q=paris+france&f=0

应用 previous bookmark EnglishTest 2018 Summer UM tools 我的桌面 - 石墨文档 NetAlgo Schedule USTC邮件系统

DBWorld location paris france search

43 results retrieved in 0.003s.

[job ann.] Professor position (tenured) in Data & Knowledge at University of Paris-Sud

2018-11-19 [release] ---- 2019-01-31 [ddl] By Bogdan Cautis

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-11/1542663138.html>

University of **Paris**-Sud, **France**. Grade: Professor...at the University of **Paris**-Sud invites applications...**Paris** area). Founded more than 35 years ago, it has over

[job ann.] PhD position on Environmental Crowd Sensing - DAVID Lab, Paris-Saclay University, Versailles, France

2018-10-29 [release] ---- 2018-11-30 [ddl] By Yehia TAHER

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-10/1540825782.html>

Webpage : [http://perso.prism.uvsq.fr/users/zeitouni/PhD\\_Proposal\\_Polluscope\\_Versailles\\_France.pdf](http://perso.prism.uvsq.fr/users/zeitouni/PhD_Proposal_Polluscope_Versailles_France.pdf)

EN-YVELINES (UVSQ) / **PARIS**-SACLAY UNIVERSITY

[job ann.] Full Funded PhD Position: Lab CReSTIC (Univ. of Reims, France), Industrial AdWanted

2018-12-10 [release] ---- 2019-01-1 [ddl] By Cyril DE RUNZ

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-12/1544445186.html>

Main place of work: **France**: **Paris** (75) or Reims (51) Type...TV, ...) online in **France**, United States and

检索会议主题 subject: web intelligence

DBWorld Demo

114.214.161.227:5000/search?p=1&q=web+intelligence&f=0

应用 previous bookmark EnglishTest 2018 Summer UM tools 我的桌面 - 石墨文档 NetAlgo Schedule USTC邮件系统

DBWorld subject web intelligence search

135 results retrieved in 0.027s.

[conf. ann.] [WIMS2019]: 9th International Conference on Web Intelligence, Mining and Semantics

2018-10-11 [release] ---- 2019-01-11 [ddl] By Khac-Hoai Nam Bui

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-10/1539274083.html>

Webpage : <http://ke.cau.ac.kr/wims2019/>

Conference on **Web Intelligence**, Mining and...Linked Semantic) **Web** Data at scale - **Intelligence** for Big Data...and the Semantic **Web** -**Intelligence** and Semantics

[conf. ann.] 2019 AAAI International Workshop on Health Intelligence (W3PHIAI 2019)

2018-09-20 [release] ---- 2018-11-5 [ddl] By Arash Shaban-Nejad

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-09/1537472298.html>

Webpage : <http://w3phiai2019.w3phi.com/>

from artificial **intelligence** and machine...**web**, and big data analytics in **web**-based healthcare applications...Support • Semantic **Web** and **Web** Services • Biomedical

[journal ann.] CFP: Elsevier FGCS special issue on Data Exploration in the Web 3.0 Age

2018-11-13 [release] ---- TBD [ddl] By Maurizio Atzori

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-11/1542133305.html>

检索发送时间sent: 2018 dec

DBWorld Demo

114.214.161.227:5000/search?q=2018+dec&f=3&p=1

应用 previous bookmark EnglishTest 2018 Summer UM tools 我的桌面 - 石墨文档 NetAlgo Scheduler USTC邮件系统

DBWorld keyword 2018 dec search

148 results retrieved in 0.012s.

[journal ann.] CFP: IP&M Special Issue on Deep Learning for Information Retrieval

2018-12-13 [release] ---- TBD [ddl] By Ben He Dr.

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-12/1544755543.html>

Webpage : <https://www.journals.elsevier.com/information-processing-and-management>

Apologies for cross-posting, ===== Information Processing & Management Special Issue on Deep Learning for Information Retrieval  
https://goo.gl/k3eziz ===== = Aims and Scope = Despite the thrilling achievements of deep neural networks on a wide range of  
tasks in computer vision and natural language processing, revolutionary improvements from deep neural IR models are yet to achieve, highlightin

[conf. ann.] CFP deadline Jan 18: HardBD&Active'19 at ICDE'19 on New Hardware + DB

2018-12-13 [release] ---- 2019-01-18 [ddl] By Shimin Chen

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-12/1544751474.html>

Webpage : <http://www.carch.ac.cn/~ictdb/HardBD-Active-2019>

Call For Papers: Joint workshop of HardBD and Active'19 co-located at ICDE 2019 in Macau SAR, China [Main focus] Exploiting new hardware technologies for data-intensive workloads and big data systems. [Important Dates] Paper submission: January 18, 2019 (Friday) 11:59:00 PM PT Notification of acceptance: February 8, 2019 (Friday) Camera-ready copies: February 22, 2019 (Friday) Workshop: April 8, 2019 (Monday) [Topics of Interest] Systems Architecture on New Hardware Data Management Issues in Software-Hardware-System Co-design Main Memory Data M

[conf. ann.] IEEE MDM 2019 Call for Demo Track Papers

2018-12-13 [release] ---- 2019-02-11 [ddl] By Baihua Zheng

检索截止时间deadline: 201902

DBWorld Demo

114.214.161.227:5000/search?q=201902&f=4&p=1

应用 previous bookmark EnglishTest 2018 Summer UM tools 我的桌面 - 石墨文档 NetAlgo Scheduler USTC邮件系统

DBWorld keyword 201902 search

56 results retrieved in 0.004s.

[conf. ann.] IEEE MDM 2019 Call for Demo Track Papers

2018-12-13 [release] ---- 2019-02-11 [ddl] By Baihua Zheng

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-12/1544748058.html>

Webpage : <https://www.comp.hkbu.edu.hk/mdm2019>

Call for Demo Track Papers Demos are an important part of MDM 2019. They provide researchers with an exciting and highly interactive way to demonstrate their cutting-edge research results and development in mobile data management. Topics of interest include, but not limited to: - Mobile Cloud Computing and Data Management in the Mobile Cloud - Data Management for Internet of Things (IoT) and Sensor Systems - Data Management for Augmented Reality Systems - Data Management for Connected Cars, Intelligent Transportation Systems, Smart Spaces - Mobile Crowd-Sourcing and Crowd-Sensing - Mobile Da

[conf. ann.] IEEE MDM 2019 Call for Demo Track Papers

2018-12-13 [release] ---- 2019-02-11 [ddl] By Baihua Zheng

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-12/1544748048.html>

Webpage : <https://www.comp.hkbu.edu.hk/mdm2019>

Call for Demo Track Papers Demos are an important part of MDM 2019. They provide researchers with an exciting and highly interactive way to demonstrate their cutting-edge research results and development in mobile data management. Topics of interest include, but not limited to: - Mobile Cloud Computing and Data Management in the Mobile Cloud - Data Management for Internet of Things (IoT) and Sensor Systems - Data Management for Augmented Reality Systems - Data Management for Connected Cars, Intelligent Transportation Systems, Smart Spaces - Mobile Crowd-Sourcing and Crowd-Sensing - Mobile Da

[conf. ann.] IEEE MDM 2019 Call for Tutorials / Advanced Seminars

2018-12-13 [release] ---- 2019-02-20 [ddl] By Baihua Zheng

## 实验总结

亮点:

1. 满足了本实验的要求, 实现了多种常用的检索目标的搜索
2. 信息准确, 搜索反应速度快, 结果展示页面清晰美观 (相关词高亮, 结果分页)

## [conf. ann.] Abstract Deadline on Dec 3:

2018-12-02 [release] ---- 2018-12-3 [ddl] By Shivnath Babu

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018->

Webpage : <http://db.cs.pitt.edu/smdb2019/>

on Self-Managing **Database** Systems [https...topics in the core datab](https://www.topicsinthearea.com/)

<< 1 2 3 4 5 6 >>

## [conf. ann.] VLDB 2019 CfP - Industrial, Applications,

2018-10-18 [release] ---- 2019-02-18 [ddl] By Wolfgang Lehner

DBWorld Message : <http://www.cs.wisc.edu/dbworld/messages/2018-10/1539873462.html>

and Streaming Data \* **Database** and Software as...a Service \* **Database** Appliances and

<< 1 2 3 4 5 6 7 8 9 10 >>

不足:

1. 爬取详细信息速度较慢 (主要原因应该在于访问网站速度慢, 打开详情网页大概需要3-4s)