Xpantrac Connection with IDEAL

Sloane Neidig, Samantha Johnson, David Cabrera, Erika Hoffman

CS 4624

5/6/2014

Project Specification

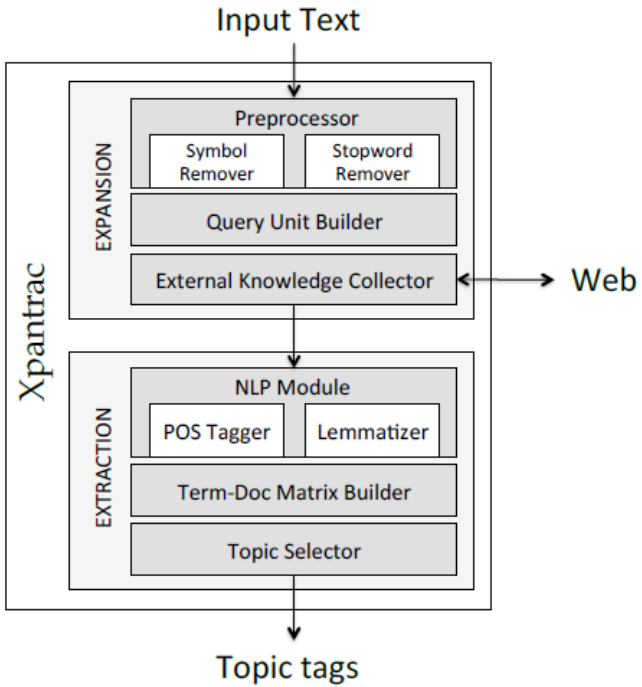
- **Short Description**
  - Integrating Xpantrac into the IDEAL software suite
  - Applying Xpantrac to identify topics for IDEAL webpages
- **Primary Contact**
  - Seungwon Yang
- **Deliverable**
  - ~~Xpantrac tailored for IDEAL~~
  - Xpantrac tailored for Solr (explained later)

What is Xpantrac?

- Seungwon Yang's dissertation topic
- Based on **Expan**sion-Ex**trac**tion approach
- Algorithm to identify topics in a given webpage
- Purpose: Tag topics to easily understand document
- Previously, only running on Seungwon's personal data set
  - 1,000 NYT articles

What is Xpantrac?

- Input: text file
- Build query
  - Every 5 words, 1 word overlap
- Send query to search API
- Find topics in retrieved documents
  - Frequency of words
- Select most frequent as “topics”
- Output: topics



Example Input/Output

Input

Knife-wielding mob kills 27 at China train station
By CNN Staff
.....
.....

Output

--------- m39 10 topics ---------
station, people, attach, news, train, china, xinhua, railway, group, knife

Progress at Midterm

- Python script written to get HTML out of WARC (web archive) files
- Xpantrac currently runs on local data set (50 .txt files) and gives correct output, using Yahoo Web API
- Became familiar with Solr and attempted to index CNN news articles
  - Stores information like title, first 30 words, last 30 words, etc. to easily use as input for Xpantrac

Progress at Midterm: Xpantrac with Yahoo Search API

- Yahoo Search API authorization
- Input: 0.txt - 50.txt (IDs in plain_text_ids.txt)

```
if query != "":
    try:
        url = ""
        if yahoo_api_type == "web":
            url = "http://yboss.yahooapis.com/ysearch/web?q=" + query
        else:
            url = "http://yboss.yahooapis.com/ysearch/news?q=" + query
        consumer = oauth2.Consumer(key=OAUTH_CONSUMER_KEY,secret=OAUTH_CONSUMER_SECRET)
        params = {
            'oauth_version': '1.0',
            'oauth_nonce': oauth2.generate_nonce(),
            'oauth_timestamp': int(time.time()),
        }

        oauth_request = oauth2.Request(method='GET', url=url, parameters=params)
        oauth_request.sign_request(oauth2.SignatureMethod_HMAC_SHA1(), consumer, None)
        oauth_header=oauth_request.to_header(realm='yahooapis.com')

        # Get search results
        http = httplib2.Http()
        resp, content = http.request(url, 'GET', headers=oauth_header)
        # print resp
        # print content
        results = simplejson.loads(content)
```

Progress at Midterm: Xpantrac with Yahoo Search API

```
PS C:\Users\sloan_000\Desktop\project> python .\Xpantrac_yahooWeb.py
1399317485.37

------------- Document ID: 0 is being processed --------------

------- m39 10 Topics ---------
station,people,attack,news,railway,xinhua,china,train,group,knife


29.3789999485 seconds
1399317514.75

------------- Document ID: 1 is being processed --------------

------- m39 10 Topics ---------
water,rain,california,weather,drought,los,angeles,storm,fire,street


81.75 seconds
1399317596.51

------------- Document ID: 2 is being processed --------------
```

Final Progress

- Indexed files into Solr
- Connected Xpantrac to Solr collection
- Evaluated extracted topics (current)

Indexing files into Solr

- ➔ Unsuccessful with IDEAL Pages script
- ➔ Manually indexed files instead

Indexing files into Solr

- Indexed 50 CNN files in XML format
- <id>, <title>, <content>

1.xml

```
1  <id>1</id>
2  <title>After forest fires and drought, now rains torment Southern
   California</title>
3  <content>Mario Vazquez grabbed his dog and got out of the way, as a
   stream of water and mud came gushing on to his streets.
4  Since California has been in the middle of its worst drought in 100
   years, it would seem that the sight of rain would be good news.
5  But mud from the streets is beginning to ooze over into yards,
   pools and houses. It has damaged two homes in Glendora so far,
   police chief Tim Staub said.</content>
```

Indexing files into Solr

- Errors
  - Wrong XML format (<field name=”id” …>)
  - Special characters (‘&’)

50docs.xml

```
<add>
<doc>
    <field name="id">0</field>
    <field name="title">Knife-wielding mob kills 27 at China train station</field>
    <field name="content">At least 27 people were killed and 109 wounded when a group of peop
    with knives stormed a railway station in the southwest Chinese city of Kunming, authoriti
    according to state news agency Xinhua.
    It was an organized, premeditated terrorist attack, authorities told the news agency. No
    been provided. A doctor with the Kunming No.1 People's Hospital told Xinhua over the phon
    not sure of the number of casualties. Xinhu said the Kunming Railway Station is one of th
    stations in southwest China.</field>
</doc>

<doc>
    <field name="id">1</field>
    <field name="title">After forest fires and drought, now rains torment Southern California
    <field name="content">Mario Vazquez grabbed his dog and got out of the way, as a stream o
```

Xpantrac for Solr

- Successfully indexed files into Solr, but 50 is too small for Xpantrac
- Use Seungwon's Wikipedia collection
  - 2.8 million documents (and counting)
- Modify Xpantrac code

Xpantrac for Solr

- ➔ Change URL to Seungwon's collection

field : query

```
for item in query_list: # [[query unit 1], [query unit 2], [query unit 3],...]
    query = "+".join(item)
    num_results_returned = 0
    if query != "":
        try:
            query_assembled = '[redacted]/solr/collection1/select?q=content%3A'+ query +'&wt=json&indent=true&rows=5';

            conn = urlopen(query_assembled)
            rsp = eval( conn.read() )
            results = rsp['response']['docs']
```

Parse document information from JSON response

Xpantrac for Solr

- Number of topics to find: 10

```
def main():
    num_topics = 10
    window_overlap = 1
```

- Number of API results to return: 10
- Query unit size: 5

```
# input text ------------------------------------ (Control inputs)
iter_id = 1
# for u_size in [20, 15, 10, 5, 1][3:4]:
for u_size in [20, 15, 10, 5, 2, 1][3:4]:       # [3:4] -> group 5 words together
    for a_return in [50, 10, 5, 1][1:2]:        # [1:2] -> ask Solr to return 10 matching documents
```

Xpantrac for Solr

- Change how ‘content’ field is handled

Running Xpantrac

0.txt

(CNN) -- Nine Ringling Bros. and Barnum and Bailey circus performers were among 11 people injured Sunday in Providence, Rhode Island, after an apparatus used in their act failed, circus spokesman Stephen Payne said.
Eight performers fell when the hair-hang apparatus -- which holds performers by their hair -- failed, Payne added. Another performer was injured on the ground, he said.
The performers were among 11 people hospitalized with injuries related to the accident, Rhode Island Hospital spokeswoman Jill Reuter told CNN. One of those people was listed in critical condition, Reuter said.
It was not immediately clear who the other two victims were.
Multiple emergency units responded to the accident at the Dunkin' Donuts Center.
Eyewitnesses told CNN affiliate WPRI that they saw acrobats up on a type of aerial scaffolding doing a "human chandelier" when a cable snapped.
Payne told CNN's Fredricka Whitfield the apparatus had been used for multiple performances each week since Ringling Bros. and Barnum & Bailey lauched its "Legends" show in February.
"Each and every time that we come to a new venue, all of the equipment that is used by this performer -- this group of performers as well as other performers -- is carefully inspected. We take the health and safety of our performers and our guests very seriously, and our company has a safety department that spends countless hours making sure that all of our equipment is indeed safe and effective for continued use," he said.
The circus and local authorities are investigating the incident together, Payne said.
"Legends" began a short Providence residency on Friday. The final five performances there were slated for 11 a.m., 3 p.m. and 7 p.m. on Sunday, and 10:30 a.m. and 7 p.m. on Monday.
"The rest of the (11 a.m. Sunday) show was canceled and we're making a determination about the remainder of the shows for the Providence engagement," Payne said.

Running Xpantrac

```
PS C:\Users\sloan_000\Desktop\project> python .\Xpantrac.py

Input text: 0.txt is being processed.........
---- List of queries (query size:5) ----

[1]: cnn+ringling+bros+barnum+bailey
[2]: bailey+circus+performers+injured+providence
[3]: providence+rhode+island+apparatus+failed
[4]: failed+circus+spokesman+stephen+payne
[5]: payne+performers+fell+hair+hang
[6]: hang+apparatus+holds+performers+hair
[7]: hair+failed+payne+performer+injured
[8]: injured+ground+performers+hospitalized+injuries
[9]: injuries+accident+rhode+island+hospital
[10]: hospital+spokeswoman+jill+reuter+told
[11]: told+cnn+listed+critical+condition
[12]: condition+reuter+clear+victims+multiple
[13]: multiple+emergency+units+responded+accident
[14]: accident+dunkin+donuts+center+eyewitnesses
[15]: eyewitnesses+told+cnn+affiliate+wpri
[16]: wpri+acrobats+type+aerial+scaffolding
[17]: scaffolding+human+chandelier+cable+snapped
[18]: snapped+payne+told+cnn+fredricka
[19]: fredricka+whitfield+apparatus+multiple+performances
```

Running Xpantrac

```
[20]: performances+week+ringling+bros+barnum
[21]: barnum+bailey+lauched+legends+time
[22]: time+venue+equipment+performer+group
[23]: group+performers+well+performers+carefully
[24]: carefully+inspected+health+safety+performers
[25]: performers+guests+seriously+company+safety
[26]: safety+department+spends+countless+making
[27]: making+equipment+safe+effective+continued
[28]: continued+circus+local+authorities+investigating
[29]: investigating+incident+payne+legends+began
[30]: began+short+providence+residency+final
[31]: final+performances+slated+rest+canceled
[32]: canceled+making+determination+remainder+providence
[33]: providence+engagement+payne

---- Micro corpus is created ----

---- Vector Space Model is applied for topic extraction ----

---- Topics (separated by ',') ----

payne, island, rhode, circus, providence, reuter, american, county, john, state

------------------------------------
```

Xpantrac with IDEAL

- IDEAL Pages group given different indexing specifications
- “Content” field contains all text in <body> of an HTML page
  - We only need relevant article text


```
{
"content": [
          "Google Newsvar GLOBAL_window=window;(function(){function
d(a){this.t={};this.tick=function(a,c,b){b=void 0!=b?b:(new
Date).getTime();this.t[a]=[b,c]};this.tick(\"start\",null,a)}var a=new
d;GLOBAL_window.jstiming={Timer:d,load:a};if(GLOBAL_window.performance&&GLOBAL_window.performance.timing){var
a=GLOBAL_window.performance.timing,c=GLOBAL_window.jstiming.load,b=a.navigationStart,a=a.responseStart;0<b&&a>=
b&&(c.tick(\"_wtsrt\",void...”],
"collection_id": "3650",
"id": "7f74825401865487f671bd0fd388ce2b",
"_version_": 1465938356823130000
},
```

Xpantrac with IDEAL

- First 30 words of “content” are not useful for our purposes

```
"Google Newsvar GLOBAL_window window function function d a
this t this tick functiona a c b b void b b new Date getTime
this t a b c this tick start null a var a new d GLOBAL_window
jstiming”
```

Xpantrac with IDEAL

As an alternative, a SOLR system could

be constructed with the following two constraints:

1. It should index a massive number of documents, so that any documents from users could

be expanded based on indexed information.

2. It should return the most relevant portion of the matching documents, for a query.

Future: Xpantrac with IDEAL

- Solr collection with IDEAL to match our specifications
  - “content” field
  - “content” only contains actual article information
- To connect that collection to Solr:

```
if query != "":
    try:
        query_assembled = '[redacted]/solr/collection1/select?q=content%3A'+ query +'&wt=json&i

        conn = urlopen(query_assembled)
        rsp = eval( conn.read() )
        results = rsp['response']['docs']
```

Evaluating Xpantrac Topics

- CTR_30, VARIOUS_30
  - Collections of 30 files each
  - Assigned topics by humans
- gold_ctr30.csv, gold_various30.csv
  - “Gold standard topics”, merged
- Evaluation Metrics
  - Precision, Recall, F1 (tag set score)
- Compare
  - computePRF1.py gold_ctr30.csv xpantrac.csv
  - computePRF1.py gold_various30.csv xpantrac.csv

References

- Automatic Identification of Topic Tags from Texts Based on Expansion-Extraction Approach by Yang, Seungwon, Virginia Polytechnic and State University, 2013, 230 pages. (Seungwon's dissertation)

Questions?