Crawling Tweets & Pre-Processing

Bayu Distiawan

Natural Language Processing & Text Mining
Short Course

Pusat Ilmu Komputer UI
22 – 26 Agustus 2016

1. Creating crawler

- Open tweepy installation folder, find streaming example
  - \installation_dir\tweepy\examples\streaming.py

2. Creating crawler (2)

- Copy your key & token from Twitter API to the code

```
# Go to http://apps.twitter.com and create an app.
# The consumer key and secret will be generated for you after
consumer_key=“”
consumer_secret=“”
# After the step above, you will be redirected to your app's page.
# Create an access token under the the "Your access token" section
access_token=“”
access_token_secret=“”
```

2. Creating crawler (3)

- This part is a listener to print received tweet to standard output for example command line

```
class StdOutListener(StreamListener):
        def on_data(self, data):
        print(data)
                return True
        def on_error(self, status):
                print(status)
```

2. Creating crawler (4)

- This part is where we put the keyword of tweets that suits our interest. From the example, we will received tweets that contains “basketball”

```
stream.filter(track=['basketball'])
```

- You can try to change with multiple keywords like
  - **stream.filter(track=[‘jokowi‘, ‘prabowo’])**
- We don’t cover the technique to enhance the keywords to make the search result better.

pusilkom ui
quality results

2. Creating crawler (5)

- Run the crawler
  - open your command line or terminal
  - change the active directory to the place of the crawler file
  - command:
    - python crawler.py
  - see what happended
  - change the command to:
    - python crawler.py > output.json

3. Preparing corpus (1)

- Filter the tweet stream, pick the attribute we want to analyze.
- In this example we only want to do some preprocessing task

4. Preparing corpus (2)

- Create new python file (transform2.py)
- Impor json, we want to read json format

```
import json

fo = open(‘file_path\output.json’, 'r')
fw = open(‘file_path\corpus.txt', 'a')
```

- fo -> read the file where the crawler produced
- fw -> create new file

pusilkom ui
quality results

4. Preparing corpus (3)

- read all line in fo
- write the tweet text to fw

```
for line in fo:
        try:
                tweet = json.loads(line)
                fw.write(tweet['text']+"\n")
        except:
                continue
```

pusilkom ui
quality results

DATA PREPROCESSING

Sentence Splitting

```
from nltk.tokenize import sent_tokenize
text = "put some free text"
sent_tokenize_list = sent_tokenize(text)
i = 1
for sent in sent_tokenize_list:
      print "Sent ", i , ": ", sent
      i+=1;
```

Stopword Removal

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

text = "put a sentence"
stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(text)

filtered_sentence = [w for w in word_tokens if not w in stop_words]

print ' '.join(filtered_sentence)
```

Stemming

```
from nltk.stem.porter import PorterStemmer
from nltk.tokenize import word_tokenize
porter_stemmer = PorterStemmer()

def stem(sent):
        words = word_tokenize(sent)
        result = list()
        for word in words:
                result.append(porter_stemmer.stem(word))
        return ' '.join(result)

print stem("David De Gea welcomed Paul Pogba 's record-
breaking return Manchester United claimed feels like Frenchman
never left Old Trafford")
```

POS Tagging

```
import nltk
sentence = "Put a sentence"
tok_sentence = nltk.word_tokenize(sentence)
tagged_sentence = nltk.pos_tag(tok_sentence)
print tagged_sentence
```

NER

```
import nltk
sentence = "Put a sentence"
tok_sentence = nltk.word_tokenize(sentence)
tagged_sentence = nltk.pos_tag(tok_sentence)
ner_sent = nltk.ne_chunk(tagged_sentence);
print ner_sent;
```

Parsing

- Download Stanford Parser
  - http://nlp.stanford.edu/software/stanford-parser-full-2015-04-20.zip
  - Extract: stanford-parser.jar, stanford-parser-3.6.0-models.jar, slf4j-api.jar

```
import os
os.environ['STANFORD_PARSER'] = 'C:/Users/MIC-UI/Desktop/stanford-parser.jar'
os.environ['STANFORD_MODELS'] = 'C:/Users/MIC-UI/Desktop/stanford-parser-3.6.0-
models.jar'

from nltk.internals import find_jars_within_path
from nltk.parse.stanford import StanfordParser
parser=StanfordParser(model_path="edu/stanford/nlp/models/lexparser/englishPCFG.ser.
gz")

results = parser.raw_parse("Mark eats noodle at night")
#print list(results)
for result in results:
          print result.draw()
```

Parsing (UNIX)

```
cd $HOME

# Update / Install NLTK
pip install -U nltk

# Download the Stanford NLP tools
wget http://nlp.stanford.edu/software/stanford-ner-2015-04-20.zip
wget http://nlp.stanford.edu/software/stanford-postagger-full-2015-04-20.zip
wget http://nlp.stanford.edu/software/stanford-parser-full-2015-04-20.zip
# Extract the zip file.
unzip stanford-ner-2015-04-20.zip
unzip stanford-parser-full-2015-04-20.zip
unzip stanford-postagger-full-2015-04-20.zip
```

Parsing (UNIX)

```
export STANFORDTOOLSDIR=$HOME

export CLASSPATH=$STANFORDTOOLSDIR/stanford-postagger-full-2015-04-20/stanford-postagger.jar:$STANFORDTOOLSDIR/stanford-ner-2015-04-20/stanford-ner.jar:$STANFORDTOOLSDIR/stanford-parser-full-2015-04-20/stanford-parser.jar:$STANFORDTOOLSDIR/stanford-parser-full-2015-04-20/stanford-parser-3.5.2-models.jar

export STANFORD_MODELS=$STANFORDTOOLSDIR/stanford-postagger-full-2015-04-20/models:$STANFORDTOOLSDIR/stanford-ner-2015-04-20/
```

Parsing (UNIX)

```
from nltk.tag.stanford import StanfordPOSTagger
st = StanfordPOSTagger('english-bidirectional-distsim.tagger')
st.tag('What is the airspeed of an unladen swallow ?'.split())

from nltk.tag import StanfordNERTagger
st = StanfordNERTagger('english.all.3class.distsim.crf.ser.gz')
st.tag('Rami Eid is studying at Stony Brook University in NY'.split())

from nltk.parse.stanford import StanfordParser
parser=StanfordParser(model_path="edu/stanford/nlp/models/lexparser/englishPCFG.ser.gz")
list(parser.raw_parse("the quick brown fox jumps over the lazy dog"))

from nltk.parse.stanford import StanfordDependencyParser
dep_parser=StanfordDependencyParser(model_path="edu/stanford/nlp/models/lexparser/englishPCFG.ser.gz")
print [parse.tree() for parse in dep_parser.raw_parse("The quick brown fox jumps over the lazy dog.")]
```