Practical Natural Language Processing with Hadoop

@DanRosanova
Senior Architect
West Monroe Partners
A little about me & West Monroe Partners

- 15 years in technology consulting
- 5 time Microsoft Integration MVP
- Author of BizTalk 2010 Patterns
- Specialize in distributed computing
- Just spoke at Big Data Tech Con (an hour ago!)

- Business & Technology Consulting
- 450+ staffers
- 10 offices across North America
- Partner of Bearing Point
Natural Language Processing

- Making computers derive meaning from human language
- Most ‘data’ that isn’t image based is natural text
- Every communication you have with every person
- There is the possibility of vast data in this text
- This is harder than it sounds
Why is language so hard?

• Volume, Variety, and Variability – sound familiar?
• Language is context sensitive – in every sense
• True language comprehension is a Strong AI / AI Complete problem
• Language kills us
Tools for NLP

- Python
- Natural Language Tool Kit (NLTK: http://www.nltk.org)
- Hortonworks Sandbox (or any Hadoop distro) http://hortonworks.com/products/hortonworks-sandbox/
- Some code from http://danrosanova.wordpress.com/nlp/ or just follow along
This letter is regarding the insurance claim for my car. My policy number is 123456789.

The details of the accident are as given below:

I parked my vehicle in the parking area at my office. Unfortunately a delivery truck tried to park between two cars and hit my car from behind. The body from behind got smashed.

When I realized I immediately contacted your customer care and gave the details. I checked all my Insurance papers and realized that I am eligible for a claim of $1000. Your Company sent a representative and filed the report and they told that they will call me soon regarding the insurance and will get the feedback from the company at the earliest.

I would like to bring to your notice that I didn’t get any correspondence from the company yet in spite of my reminders for last ten days.

Kindly look into it and expecting a positive response at the earliest.

Thanking you
The NLP Process

- Segmentation
- Tokenization
- PoS Tagging
- Entity Detection
- Relation Detection

Breaking Into Sentences
Breaking Into Words
Part of Speech
Chunking based on PoS
Relation between Entities
Segmentation

• Need to know where to break (\n really means nothing)

• Period isn’t always a period Dr. Brown
  • Perhaps unless you have an M.D.

• Segmentation is quite tricky - NLTK includes a good sentence segmenter

• You can also try using your own – I wouldn’t
Tokenization

• Breaking the sentence into words and punctuation

• This is our second step – process each line in the letter in a way that makes sense.
  
  raw = open('AutoClaimLetter.txt').read()
  tokens = nltk.word_tokenize(raw)

• ['This', 'is', 'regarding', 'the', 'insurance', 'claim', 'for', 'my', 'car.', 'My', 'policy', 'number', 'is', '123456789.', 'The', 'details', 'of', 'the', 'accident', 'are', 'as', 'given', 'below', ':', 'I']
Part of Speech Tagging

- nltk.pos_tag(tokens)
- [('This', 'DT'), ('is', 'VBZ'), ('regarding', 'VBG'), ('the', 'DT'), ('insurance', 'NN'), ('claim', 'NN'), ('for', 'IN'), ('my', 'PRP$'), ('car.', 'NNP'), ('My', 'NNP'), ('policy', 'NN'), ('number', 'NN'), ('is', 'VBZ')]

- Often N-Gram tagging works best

- I bet 5th grade English makes a lot more sense now!
Entity Detection

- Called Chunking
- Done with tags or trees
- Better to avoid trees if possible for Map Reduce later
Relation Detection

- [Named Entity] [some words between] [Named Entity]
- [Dan] *went for a walk with his dog* [Seamus]
Introduction to Hadoop

• Self managing & self healing
• Scale Linearly
• Programs go to data – NOT the normal way
• Simple core – modular and extensible
• It’s a file system – think of basic I/O operations
The Hadoop Ecosystem

Applications Run Natively **IN** Hadoop

- **BATCH** (MapReduce)
- **INTERACTIVE** (Tez)
- **ONLINE** (HBase)
- **STREAMING** (Storm, S4,...)
- **GRAPH** (Giraph)
- **IN-MEMORY** (Spark)
- **HPC MPI** (OpenMPI)
- **OTHER** (Search) (Weave...)

**YARN** (Cluster Resource Management)

**HDFS2** (Redundant, Reliable Storage)
Why Map Reduce Streaming?

- Batch Based - Map Reduce
- Command line – which was made for text
- Lowest common approach / works with anything
- Sends Key Value pairs between steps
  - Provides compose-ability
- Easily processes all files in directories
If you’re following along on Sandbox

- You need to run some commands to make all this work
  
  ```
  yum install numpy
  easy_install -U distribute
  pip install -U pyyaml nltk
  python
  >>>import nltk
  >>>nltk.download()
  >>>book
  ```

- Or within your program
  
  ```
  nltk.download('maxent_treebank_pos_tagger')
  ```
Basic Word Count in Python

- Requires
  - Mapper (python script)
  - Reducer (python script)
  - Input text (in HDFS)

- Easy sort of “Hello World” for Hadoop / Map Reduce

- Started from the bash shell (though can be started from the Job Designer)
Word Count Mapper (wcmap.py)

```python
#!/usr/bin/python
import sys

for line in sys.stdin:
    line = line.strip()
    words = line.split()
    for word in words:
        print '%s\t%s' % (word, 1)
```
Word Count Reducer (wcreduce.py)

```python
#!/usr/bin/python
from operator import itemgetter
import sys

echo "jim bob dan bob jim jon" | python wcmap.py | sort -k1,1 | python wcreduce.py

cur_word = None
cur_count = 0
word = None
for line in sys.stdin:
    line = line.strip()
    word, count = line.split(' \t', 1)
    count = int(count)
    if cur_word == word:
        cur_count += count
    else:
        if cur_word:
            print '%s \t %s' % (cur_word, cur_count)
        cur_count = count
        cur_word = word

if cur_word == word:
    print '%s \t %s' % (cur_word, cur_count)
```
Testing this before Hadoop

"jim bob dan bob jim jon" | python wcmap.py | sort -k1,1 | python wcreduce.py

#Outputs

bob 2
dan 1
jim 2
jon 1

#And with real text

cat AutoClaimLetter.txt | python wcmap.py | sort -k1,1 | python wcreduce.py
Running on Hadoop


Executable: hadoop

- Parameter:
  - jar /usr/lib/hadoop-mapreduce/hadoop-streaming-2*.jar
  - file 'wcreduce.py'
  - file 'wcmap.py'
  - input /NLP/Data/Ch2.txt
  - output /NLP/output
  - mapper 'python ./wcmap.py'
  - reducer 'python ./wcreduce.py'
  - numReduceTasks 2
What we have?

• Perhaps the most useless NLP program in the world!

• Why is word count so poor? Examples:
  • “That album is sick” “Those oysters made me sick”
  • “I am was happy to see them” “I was not happy to see them”

• Again – context means everything
How does real NLP look on Hadoop

- Each input and output is a file in HDFS
- The first input (raw text) may be a whole directory of files
- Cleaned text is stripped / trimmed and normalized (lower case)
- PoS tagged lists are saved
Everything to PoS in one map!

#!/usr/bin/python

import sys
import nltk, re, pprint

linenum = 0
#for each line clean and tag
for line in sys.stdin:
    line = line.strip()
    line = line.lower()
    text = nltk.word_tokenize(line)
    print '%s\t%s' % (linenum, nltk.pos_tag(text))
    linenum += 1
Some real world examples

• Clinical medical data
• Medical insurance claims
• Auto insurance claims
• Sentiment analysis
• Fraud detection
• How do you do all of these?
Some real world examples

• Here we’d want the Diagnosis to stay intact

• We would definitely need to use a Medical specific corpus / vocabulary

• There is some good categorization already
  • Title
  • Pathology Report

• This type of record is better suited to research or decision support

Practical Natural Language Processing with Hadoop

@DanRosanova
Senior Architect
West Monroe Partners