CLASSIFICATION

CS5604 Information Storage and Retrieval - Fall 2016
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

Professor: E. Fox

Presenters:
Saurabh Chakravarty,
Eric Williamson

December 1, 2016
TABLE OF CONTENTS

• Problem Definition
• High-Level Architecture
• Data-Retrieval and Processing
• Classification
• Experimental Results
• Conclusion and Future work
PROBLEM STATEMENT

• Given the tweets in the GETAR and IDEAL collections and a set of real world events, determine which tweets belong to each real world event.
PROBLEM STATEMENT

Tweet Collection

- 20: hurricane sandy
- 27: hurricane
- 188: #Arthur
- 182: #tornado
- 632: fairdale
- 174: #Manhattan

Real world event

- Hurricane Sandy
- Hurricane Arthur
- Fairdale Tornado
- Manhattan Explosion
HIGH LEVEL ARCHITECTURE

Classification pipeline

Apache HBase

Hadoop HDFS
TRAINING PIPELINE

Training data
TRAINING PIPELINE

Pre-processing
(clean/lemmatize/remove stop words)
TRAINING PIPELINE

Pre-processing
(clean/lemmatize/remove stop words)

Train word vectors
TRAINING PIPELINE

- Training data

  Pre-processing
  (clean/lemmatize/remove stop words)

  Train word vectors

  Word2Vec Model
TRAINING PIPELINE

Pre-processing
(clean/lemmatize/remove stop words)

Train word vectors

Train Classifier

Word2Vec Model
TRAINING PIPELINE

Pre-processing
(clean/lemmatize/remove stop words)

Train word vectors

Train Classifier

Training data

Word2Vec Model

Classifier Model
PREDICTION PIPELINE

Pre-processing
(clean/lemmatize/remove stop words)
PREDICTION PIPELINE

Pre-processing
(clean/lemmatize/remove stop words)

Load Word2Vec & Classifier model

Word2Vec Model
Classifier Model
PREDICTION PIPELINE

Pre-processing
(clean/lemmatize/remove stop words)

Load Word2Vec & Classifier model

Predict Tweet Event

<<table>>
ideal-
cs5604f16

[APACHE HBASE]

[Word2Vec Model]
[Classifier Model]
PREDICTION PIPELINE

Pre-processing
(clean/lemmatize/remove stop words)

Load Word2Vec & Classifier model

Predict Tweet Event

<<table>>
ideal-
cs5604f16

Word2Vec Model
Classifier Model
DATA RETRIEVAL CHALLENGES

- Large amounts of data (1.5 billion tweets)
- Have to avoid serial execution
- Cannot fit into memory
- Prevent reprocessing data unnecessarily
# DATA RETRIEVAL FROM HBASE

<table>
<thead>
<tr>
<th>Retrieval Method</th>
<th>Description</th>
<th>Smaller collection performance</th>
<th>Larger collection performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spark HadoopRDD</td>
<td>Load data into driver and parallelize across the cluster.</td>
<td>Seamless reading from HBase.</td>
<td>Hangs and does not complete reading on collections &gt; one million.</td>
</tr>
<tr>
<td>Batch Processing</td>
<td>Load one batch at a time onto the drive and parallelize across the cluster.</td>
<td>Slower reading due to batch overhead.</td>
<td>Allows classification or arbitrarily large collections.</td>
</tr>
</tbody>
</table>
CHALLENGES WITH TWEETS

- Abbreviations and Slang (u, omg)
- Non-English URLs and characters
- Misspellings

RT: @AssociationsNow A Year After Texas Explosion Federal Repourt Outlines Progress on Fertilize...
http://t.co/8fDbMu9asU #meetingprofs
TEXT-PREPROCESSING

• Remove URLs
• Remove the # characters
• Lemmatization using StanfordNLP
• Stopword removal
RT: @AssociationsNow A Year After Texas Explosion Federal Report Outlines Progress on Fertilize...
http://t.co/8fDbMu9asU #meetingprofs

year texas explosion federal report outline progress fertilize
meetingprof
TRAINING DATA GENERATION

• Random samples of collections corresponding to real world events.

• Tweet assigned a number with the class it belongs to most.

• Hand Labeled 1000 Tweets
CLASS DISTRIBUTION FOR TRAINING AND TEST DATA

Class distribution of manually labeled data

Number of Tweets

Class Number

Class distribution of manually labeled data
TEXT CLASSIFICATION

- Feature selection
- Feature representation
- Choice of classifier
# A Comparison of Feature Selection Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tf-idf</td>
<td>• Superior for small feature.</td>
<td>Accuracy suffers for large datasets.</td>
</tr>
<tr>
<td></td>
<td>• High term removal capability.</td>
<td></td>
</tr>
<tr>
<td>Mutual information</td>
<td>Simple to implement.</td>
<td>Inferior accuracy performance.</td>
</tr>
<tr>
<td>Association rules</td>
<td>• Fast execution.</td>
<td>Prone to discovering too many rules or poorly understandable rules that hurt performance and interpretation.</td>
</tr>
<tr>
<td></td>
<td>• Very good accuracy for multi-class scenarios.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Easy to interpret the rules</td>
<td></td>
</tr>
<tr>
<td>Chi-square statistic</td>
<td>Robust accuracy and performance with large sample sets with fewer classes.</td>
<td>Difficulty in interpretation of when there are a large number of classes.</td>
</tr>
<tr>
<td>Within class popularity</td>
<td>Identifies words that are most discriminative.</td>
<td>Ignores the sequence of words.</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>Captures relationships of a word with neighbors.</td>
<td>High computational complexity. Long training time for large sample size.</td>
</tr>
</tbody>
</table>

For more details, please refer the appendix section.
COMMON FEATURE REPRESENTATION TECHNIQUES

- One-hot encoding
- Bag of words

Challenges
- Large number of dimensions
- Word relationships with neighbors are not captured
WORD2VEC

• A feature selection technique.
• Captures the semantic context of a word’s relation with neighbors.
Similar words are grouped together and closer to one another.

Word displacements are relationships between the words.

Source: *Linguistic Regularities in Continuous Space Word Representations*, Mikolov et al, 2013
**WORD2VEC**

![Diagram of Word2Vec model](http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/)
WORD2VEC

Hidden Layer Weight Matrix

300 neurons

10,000 words

Word Vector Lookup Table!

300 features

10,000 words

Slide courtesy - http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
CLASSIFIER – IMPLEMENTATION DETAILS

• A word feature is an average of the word vectors generated by the Word2Vec model.
• We used a vector representation with a default of 100 values.
• We chose the multi-class logistic regression in the Spark framework to perform classification.
• The classifier labels the predicted class along with the normalized probabilities of the other classes.
EXPERIMENTS

• Effect of preprocessing
• Accuracy performance
• Runtime performance
• Probability distribution
• Class assignment
CLEANING EXPERIMENT

• Determine how cleaning the data influences accuracy

• Cleaning:
  • Lemmatization
  • Stopword removal
  • Hashtag removal

• Experimental setup:
  • Split hand-labeled data:
    • 70% train
    • 30% test
CLEANING IMPROVES ACCURACY!

Training and Testing Data

- Without Cleaning
- With Cleaning

<table>
<thead>
<tr>
<th>Method</th>
<th>Without Cleaning</th>
<th>With Cleaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word2Vec with logistic regression</td>
<td>0.84</td>
<td>0.96</td>
</tr>
<tr>
<td>Association Rules</td>
<td>0.88</td>
<td>0.94</td>
</tr>
</tbody>
</table>

- Word2Vec with logistic regression: 29% fewer misclassifications
- Association Rules: 51% fewer misclassifications
ACCURACY EXPERIMENT

• Determine which classifier gives better results on labeled data

• Experimental setup:
  • Generate 10 different breakups of the labeled data
  • Calculate metrics for each classifier on same breakup
  • 9 different classes
Labeled Data Divided into 10 sets

Training 7 sets

Rotate sets

Test 3 sets

Generate 10 different training and test sets

ACCURACY EXPERIMENTAL SETUP
Word2Vec with logistic regression had a 6.7% increase in F1 score over association rules.
CLASSIFIER RUNTIME PERFORMANCE

• Need to be able to handle large collections efficiently to classify all of the tweets

• Classify at a rate faster than the tweets coming in

• Allow reruns as more classes are added to the training set
CLASSIFICATION RUNTIME PERFORMANCE

Classifier prediction performance

Seconds to predict

Number of tweets to predict

Word2Vec with logistic regression
Association Rules
OPTIMIZATION

Broadcast models to each partition

Word2Vec Model

Logistic Regression Model

Partition 1
- Clean
- Classify
- Write

Partition 2
- Clean
- Classify
- Write

Partition 3
- Clean
- Classify
- Write

HBase Read

Partition and distribute
PROCESSING ACROSS PARTITIONS INCREASES RUNTIME PERFORMANCE!

- Word2Vec with logistic regression
- Association Rules
- Optimized Word2Vec

57% faster than original Word2Vec
14% faster than Association Rules
PROBABILITY DISTRIBUTION FOR TEST DATA
MULTI-CLASS ASSIGNMENT DISTRIBUTION FOR TEST DATA

- Class 1: 23% (1)
- Class 2: 35% (2)
- Class 3: 29% (3)
- Class 4: 13% (4)
CONCLUSION

• Reading data in blocks from HBase and then partitioning it into parallel tasks results in huge run time performance efficiency and predictability.

• Cleaning text based on the English usage nuances in the Twitter universe results in better accuracy.

• Feature selection methods like Word2Vec that capture richer word semantics and context result in better accuracy than traditional ones for text classification.

• It is natural for a Tweet to be classified in multiple classes and the tradeoff between precision and recall is dependent on the user/product requirements.
FUTURE WORK

• The system can be retrained using a bigger corpus to generate a newer set of word vectors. Training on a text corpus like Google News can help generate word vectors that have richer word relationships encoded within. These can help improve the classification accuracy.

• The Logistic Regression classifier can be retrained on new classes.

• The system will be configured to run via a cron job periodically.

• In addition to classifying a tweet, the system also emits probabilities of all the classes that could be saved in HBase and can be used by SOLR or the front-end team to use as a criterion for customizing the indexing or user experience.

• Comparisons can be performed with the results of the developed classifier with the AR classifier or a few more classifiers and an inter-classifier agreement analysis can throw further light on the efficacy of the developed classifier.
ACKNOWLEDGEMENTS

We would like to acknowledge and thank the following for assisting and supporting us throughout this project.

• Dr. Edward Fox, Dr. Denilson Alves Pereira
• NSF grant IIS - 1619028, III: Small: Collaborative Research: Global Event and Trend Archive Research (GETAR)
• NSF grant IIS - 1319578, III: Small: Integrated Digital Event Archiving and Library (IDEAL)
• Digital Library Research Laboratory
• Graduate Research Assistant – Sunshin Lee
• Other teams in CS 5604
KEEP CALM
PRESENTATION IS OVER
ANY QUESTIONS?
<table>
<thead>
<tr>
<th>Technique</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tf-idf</td>
<td>Superior for small feature sets that have a large scatter of features among the classes. High term removal capability.</td>
<td>Accuracy suffers for large data sets where a term distribution alone does not suffice in class discrimination.</td>
</tr>
<tr>
<td>Mutual information</td>
<td>Simple to implement.</td>
<td>Inferior performance in estimation of probabilities because of bias.</td>
</tr>
<tr>
<td>Association rules</td>
<td>Fast execution. Very good accuracy for multi-class scenarios. Rule based classifier helps understand the classification decision easily.</td>
<td>Prone to discovering too many rules or poorly understandable rules that hurt performance and interpretation.</td>
</tr>
<tr>
<td>Chi-square statistic</td>
<td>Robust accuracy and performance with large sample sets with fewer classes.</td>
<td>Difficulty in interpretation of when there are a large number of classes.</td>
</tr>
<tr>
<td>Within class popularity</td>
<td>Identifies words that are most discriminative.</td>
<td>Ignores the sequence of words.</td>
</tr>
<tr>
<td>Word2Vec</td>
<td>Captures relationships of a word with neighbors.</td>
<td>High computation complexity. Long training time for large sample size.</td>
</tr>
</tbody>
</table>
### WORD2VEC

**INPUT TEXT:**

```
Once upon a time in a land far away...
```

**Table: Sample and Context**

<table>
<thead>
<tr>
<th>Sample #</th>
<th>$w_0$</th>
<th>Context $C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>once...</td>
<td>{upon, a}</td>
</tr>
<tr>
<td>4</td>
<td>time...</td>
<td>{upon, a, in, a}</td>
</tr>
</tbody>
</table>

Slide courtesy - [http://files.meetup.com/12426342/5_An_overview_of_word2vec.pdf](http://files.meetup.com/12426342/5_An_overview_of_word2vec.pdf)
WORD2VEC

- Can learn the word vectors via two forms.

- **CBOW**

  Given only the current context \( C \), e.g.
  \[
  C = \{\text{upon, a, in, a}\}
  \]
  predict which of all possible words is the current word \( w_0 \), e.g. \( w_0 = \text{time} \).

  Predict the word, given the context.

Slide courtesy - [http://files.meetup.com/12426342/5_An_overview_of_word2vec.pdf](http://files.meetup.com/12426342/5_An_overview_of_word2vec.pdf)
WORD2VEC

• Skip-gram – Inverse objective of CBOW. Predict the context, given a word.

REAL WORLD EVENTS USED FOR EXPERIMENTS

The real world event along with the amount of tweets labeled as that class for our experimental sets.

<table>
<thead>
<tr>
<th>Event Description</th>
<th>Tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hurricane Sandy</td>
<td>108</td>
</tr>
<tr>
<td>Hurricane Isaac</td>
<td>83</td>
</tr>
<tr>
<td>New York Firefighter Shooting</td>
<td>58</td>
</tr>
<tr>
<td>Kentucky Accidental Child Shooting</td>
<td>16</td>
</tr>
<tr>
<td>Newtown School Shooting</td>
<td>157</td>
</tr>
<tr>
<td>Manhattan Building Explosion</td>
<td>189</td>
</tr>
<tr>
<td>China Factory Explosion</td>
<td>178</td>
</tr>
<tr>
<td>Texas Fertilizer Explosion</td>
<td>120</td>
</tr>
<tr>
<td>Hurricane Arthur</td>
<td>169</td>
</tr>
</tbody>
</table>
The real world events classified along with some collections tweets of that event are found.

<table>
<thead>
<tr>
<th>Real World Event</th>
<th>Collections</th>
<th>Real World Event</th>
<th>Collections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hurricane Sandy</td>
<td>23,27,375</td>
<td>Hurricane Isaac</td>
<td>27,28,375</td>
</tr>
<tr>
<td>New York Firefighter Shooting</td>
<td>43,46</td>
<td>Kentucky Accidental Child Shooting</td>
<td>45,46</td>
</tr>
<tr>
<td>Newtown School Shooting</td>
<td>41,42,46</td>
<td>Manhattan Building Explosion</td>
<td>173,174,399,400</td>
</tr>
<tr>
<td>China Factory Explosion</td>
<td>231,232</td>
<td>Texas Fertilizer Explosion</td>
<td>77,381</td>
</tr>
<tr>
<td>Hurricane Arthur</td>
<td>27,187,188,375</td>
<td>Quebec Train Derailment</td>
<td>96,98,381</td>
</tr>
<tr>
<td>Fairdale Tornado</td>
<td>406,632</td>
<td>Oklahoma Tornado</td>
<td>406,84</td>
</tr>
<tr>
<td>Mississippi Tornado</td>
<td>406,528</td>
<td>Alabama Tornado</td>
<td>406,407</td>
</tr>
</tbody>
</table>