Text Classification

CS5604 Information Retrieval and Storage – Spring 2016

Virginia Polytechnic Institute and State University

Blacksburg, VA

Professor: E. Fox

Presenters:
Hossameldin Shahin
Matthew Bock
Michael Cantrell

May 3rd, 2016

Agenda

- Background
- Problem Statement
- Requirements and Design
- Implementation Details
- Evaluation
- Conclusion and future work

Background

- Classification is the process of determining which predefined class or set of classes a document can be sorted into.

- In our case:
  - Classes based on IDEAL project collections
  - Binary classification (relevant or nonrelevant to the collection)

Problem Statement

- For every tweet in a collection of tweets, determine the likelihood that tweet is actually relevant to the collection
- Twitter hashtags do a good job of naturally filtering tweets
  - Still a significant amount of spam or otherwise irrelevant tweets
- Write this probability to a column in HBase for use by other teams in their systems

Problem Statement

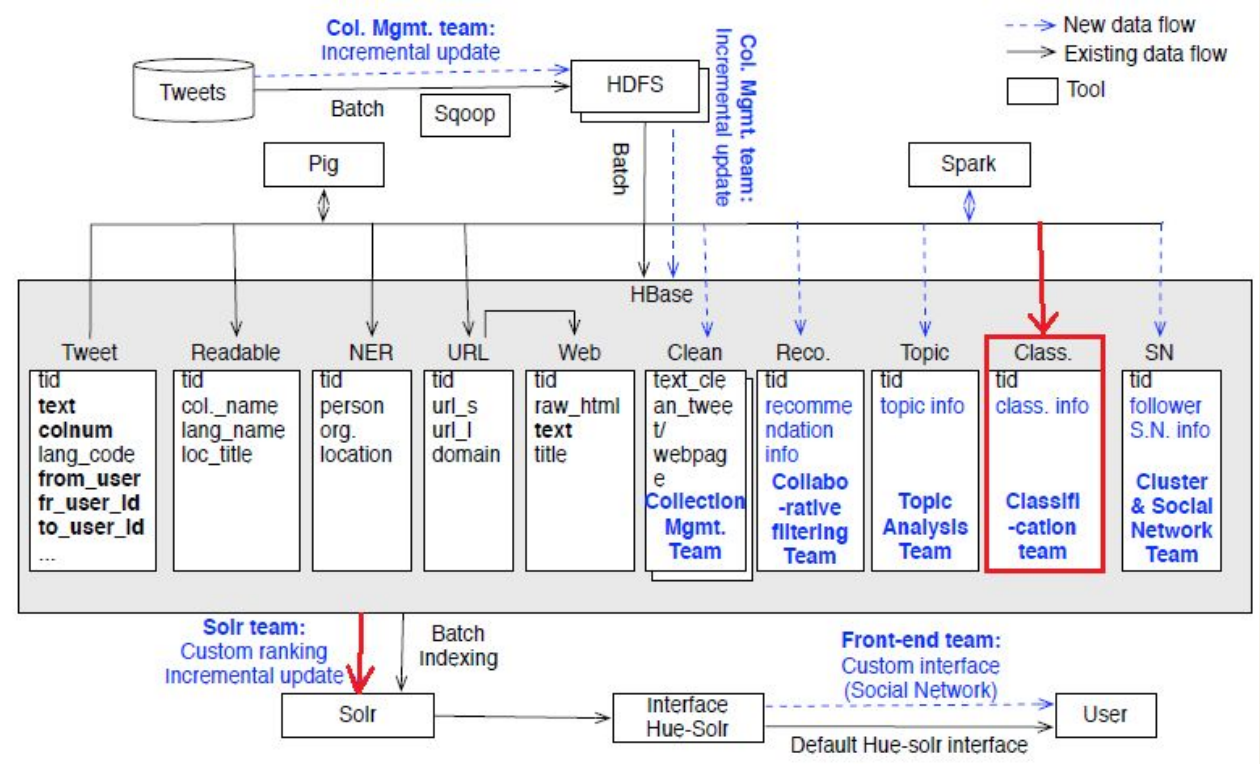
@JoelOsteen: There will be storms in life but they are only temporary. Praying for our friends in Houston and Texas. #HoustonFlood

Relevance probability: ~0.92

*#houstonflood #cambodia #GetStupid Watched a video *Car insurance quotes online* GREAT! Here: https://t.co/zChGDwQ8g9 http://t.co/nmq8Eiq8Np*

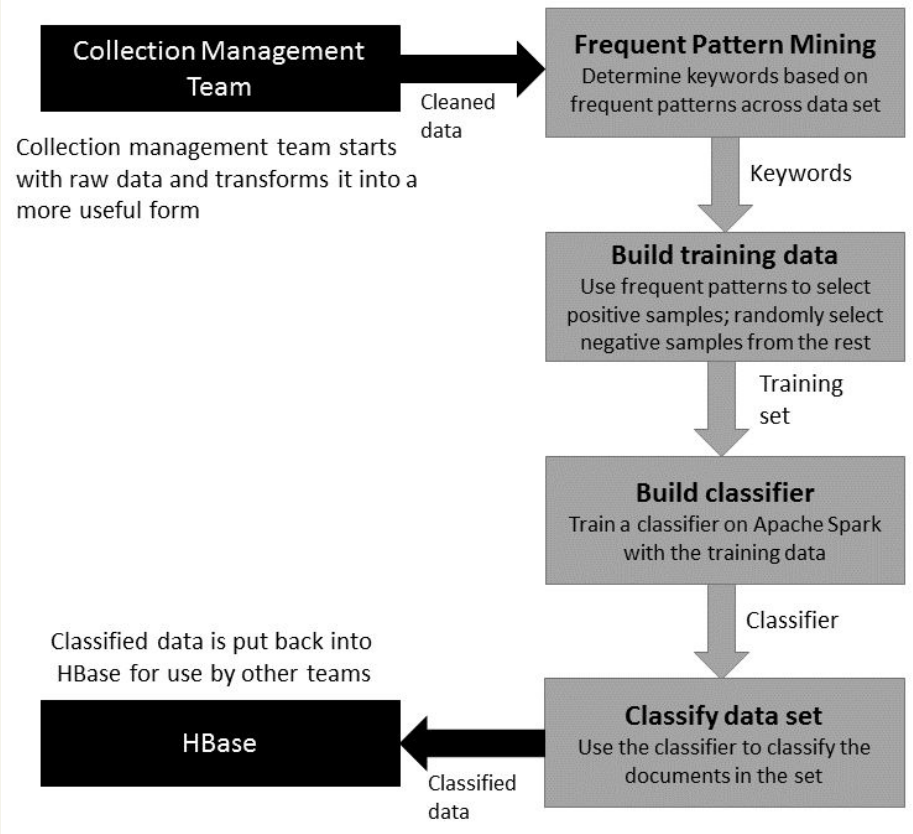
Relevance probability: ~0.08

Problem Statement



Requirements and Design

1. Feed cleaned text from Collection Management team into FPM process
2. Run tweet data through our process (described in more detail later)
3. Write probabilities back to `Classification:Relevance` column HBase for other teams to use



Implementation Details

- Spark v1.6
  - IPython notebook
  - Cluster vs. VM (Thanks to IT Security Lab at VT)
- Training data preparation
- End-To-End workflow
- Logistic regression classifier

Prepare Training data

- To train a binary classifier we need to prepare a training data
  - Manual labeling: Requires a huge work
  - Semi-Automated
    - Query Solr for positive data set
    - ***Use Frequent Pattern Mining to discover frequent itemsets***
  - Automated: Not feasible

Frequent Pattern Mining

- FPM is an algorithm for frequent itemset mining and association rule learning over transactional databases.
- FPM has two steps:
  - Identify the frequent individual items
  - Extend them to larger and larger item sets as long as those item sets appear sufficiently often in the database (minimum support)

Frequent Pattern Mining (Example)

Itemsets
{1,2,3,4}
{1,2,4}
{1,2}
{2,3,4}
{2,3}
{3,4}
{2,4}

→

Item	Support
{1}	3
{2}	6
{3}	4
{4}	5

→

Item	Support
{1,2}	3
{1,3}	1
{1,4}	2
{2,3}	3
{2,4}	4
{3,4}	3

→

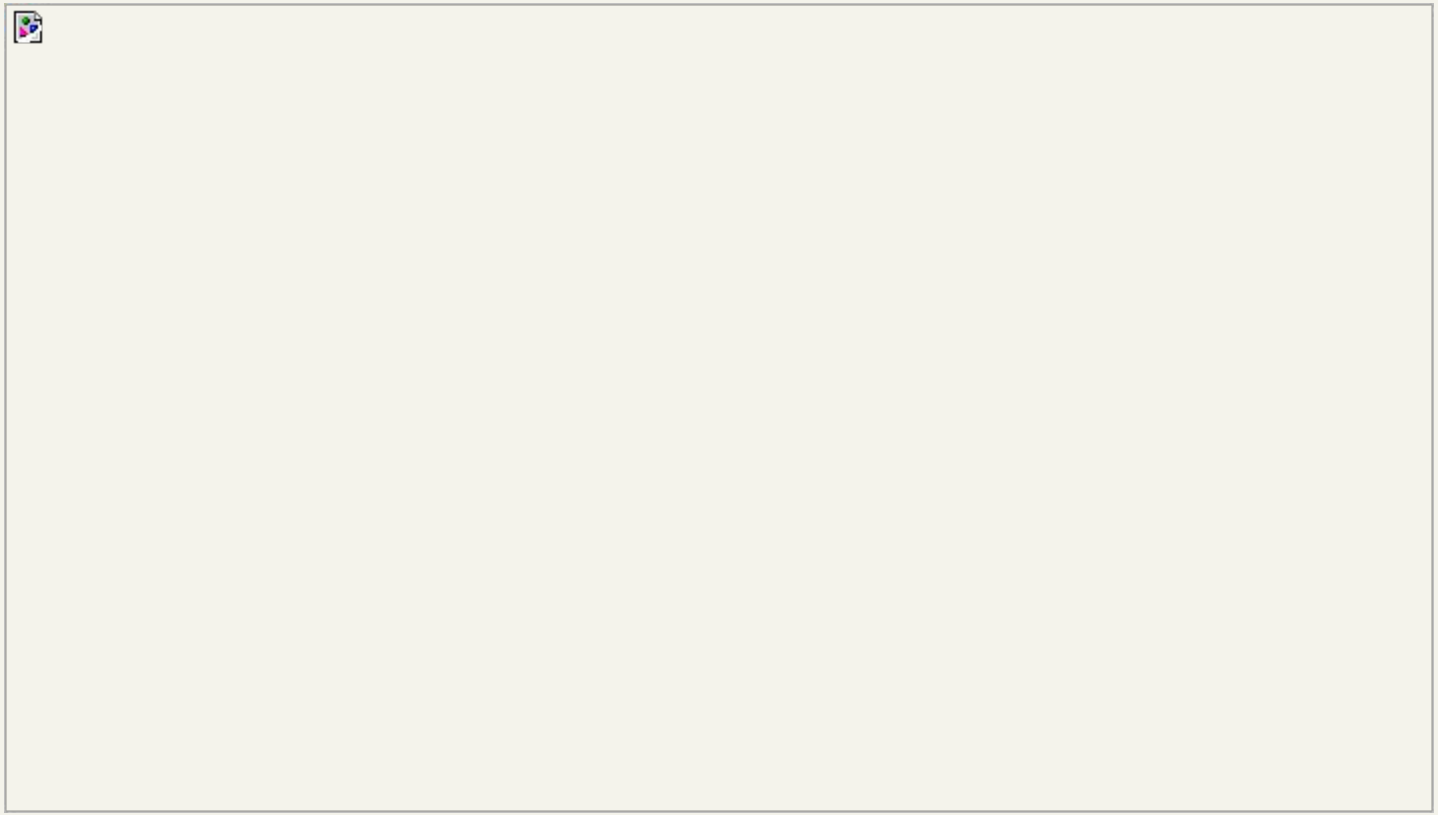
Item	Support
{2,3,4}	2

Minimum Support = 3

Frequent Pattern Mining (#GermanWings)

```
183  5664 african stood lives charliehebdo world
184  5664 african stood lives charliehebdo germanwings
185  5664 african stood lives charliehebdo
186  5664 african stood lives
187  5664 african stood germanwings
188  5664 african stood charliehebdo world germanwings
189  5664 african stood charliehebdo world
190  5664 african stood charliehebdo germanwings
191  5664 african stood charliehebdo
192  5664 african stood
193  5664 african matter lives germanwings
194  5664 african matter lives
195  5664 african matter germanwings
196  5664 african matter
197  5664 african lives world germanwings
198  5664 african lives world
199  5664 african lives charliehebdo world germanwings
200  5664 african lives charliehebdo world
201  5664 african lives charliehebdo germanwings
202  5664 african lives charliehebdo
203  5664 african charliehebdo world germanwings
204  5664 african charliehebdo world
205  5664 african charliehebdo germanwings
```

Workflow



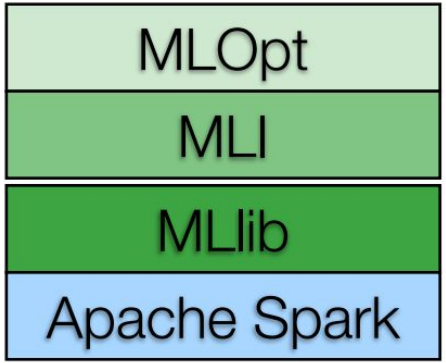
Which Classifier to use?

How much training data do you have?

None	• Manually written rules • Amount of work required is huge • Hand crafting rules produce high accuracy
Very little	• High bias models (e.g. Naïve Bayes) • Semi-supervised training methods (Boosting, EM) (Ch-16)
Reasonable	• ***Best situation!*** • ***Can use any model***
Huge	• The most accurate results • Expensive and impractical (SVMs train time or kNN test time) • Naïve Bayes might become the best choice again!

Why Logistic Regression?

Current Spark Architecture:



MLOpt: Autotuners for ML pipelines

MLI: Experimental API to simplify ML development

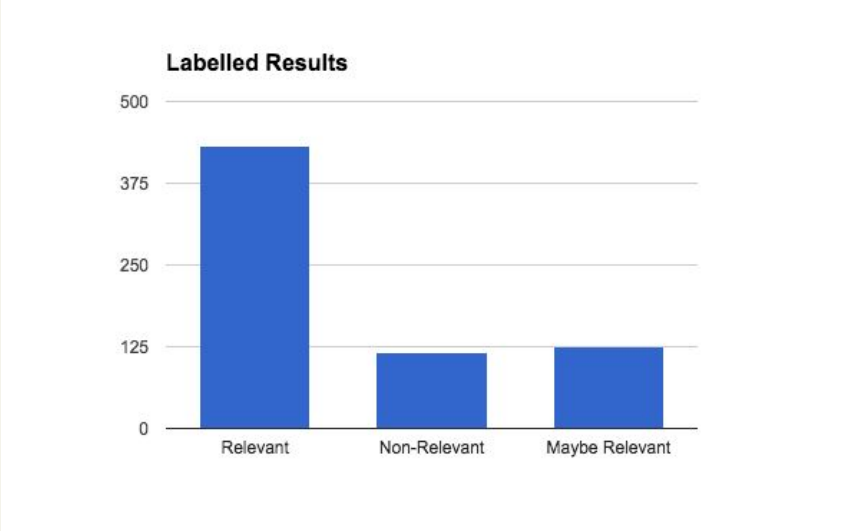
MLlib: Spark's core ML library

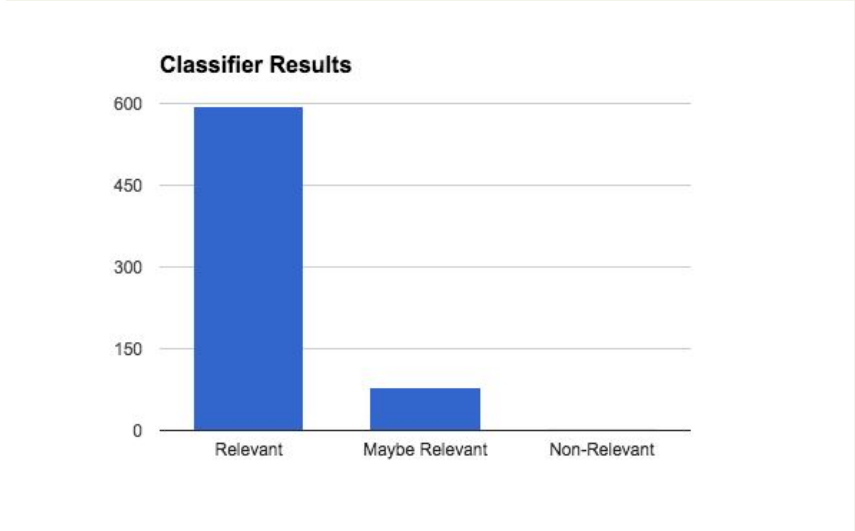
Evaluation

- Constructed 7 Surveys for manual labelling
  - 675 tweets labelled from wdbj7shooting collection
  - Ideally 3 responses per 100 tweets

- 16/21 responded to surveys
  - All but one survey had multiple responses
  - Some didn’t label all tweets on form

Survey Results





Classifier marks more as Relevant
than labelled results

Very few marked as Non-Relevant

Conclusion and Future Work

- Evaluation shows that classifier accuracy could be improved, but trends similarly to manually labelled results
  - More structured evaluation should be performed to validate results of initial evaluation
  - Relevance thresholds may need to be determined for each collection
- Our design will make it easy to apply our system to other collections in the future
- First priority for future work is to put more time into web page processing
  - Our system should cover web pages with minimal modifications, but we have not done much testing

Acknowledgments

- NSF grant IIS - 1319578, III: Small: Integrated Digital Event Archiving and Library (IDEAL).

- Dr. Fox

- GRAs
  - Mohamed Magdy Farag
  - Sunshin Lee

- Other teams

Thank you!

Questions?