News Classifier and Trending Event Detection

https://github.com/sujithcp/NewsAnalyser

SHARANYA KV

SUJITH CP

A Brief Introduction

Data classification and mining are trending and rapidly evolving fields of Computer science. Data mining is the process of identifying patterns and relationships in data that often are not obvious in large and complex data sets.
News event detection and classification are very helpful in socio-economic research and such purposes. As social media and real-time information sharing gains popularity, an automated event detection system can do better in understanding the required events.

Machine Learning Technologies for Data Mining

- Inductive Logic Programming
- Genetic Algorithms
- Neural Networks
- **Statistical Methods (We use Bayesian method for classification)**
- Decision Trees
- Hidden Markov Models

Procedure for building news classifier

Data Collection

- RSS links collection
- Feed fetching

Preprocessing and cleaning

- Extract news from feed links using beautifulsoup and sumy
- Discard very short and irrelevant data

Transformation and reduction

- Tokenizing,stopwords removal, stemming etc.
- Prepare categorized datasets for training

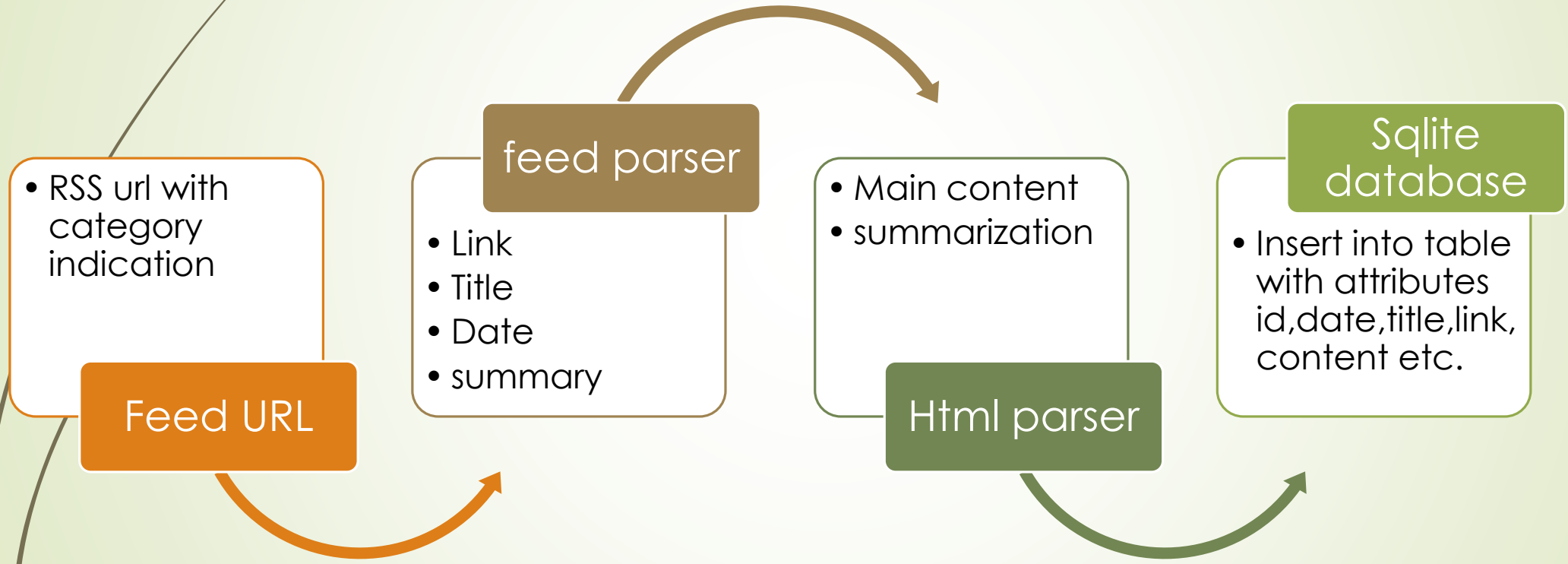
Training

- Train the machine using naïve Bayes classifier.(Used nltk module for the same)
- Dump the classifier object for easy retrieval.(Using pickle)

Testing

- Test 1: Use same dataset for training and classification.
  - Expecting higher accuracy.
- Test 2: Use mutually exclusive 50% data for training and classification.

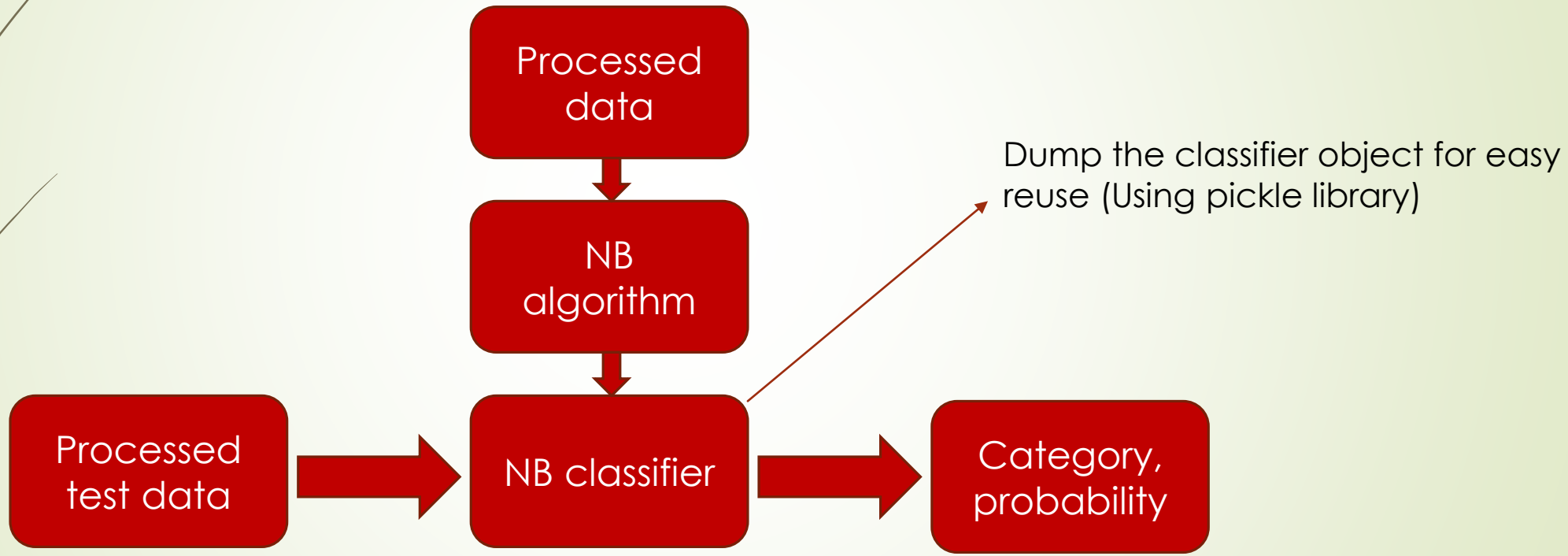
Data collection and cleaning



Training

- Supervised learning using naïve Bayes classifier

  NB classifier works on the Bayes theorem in probability mathematics

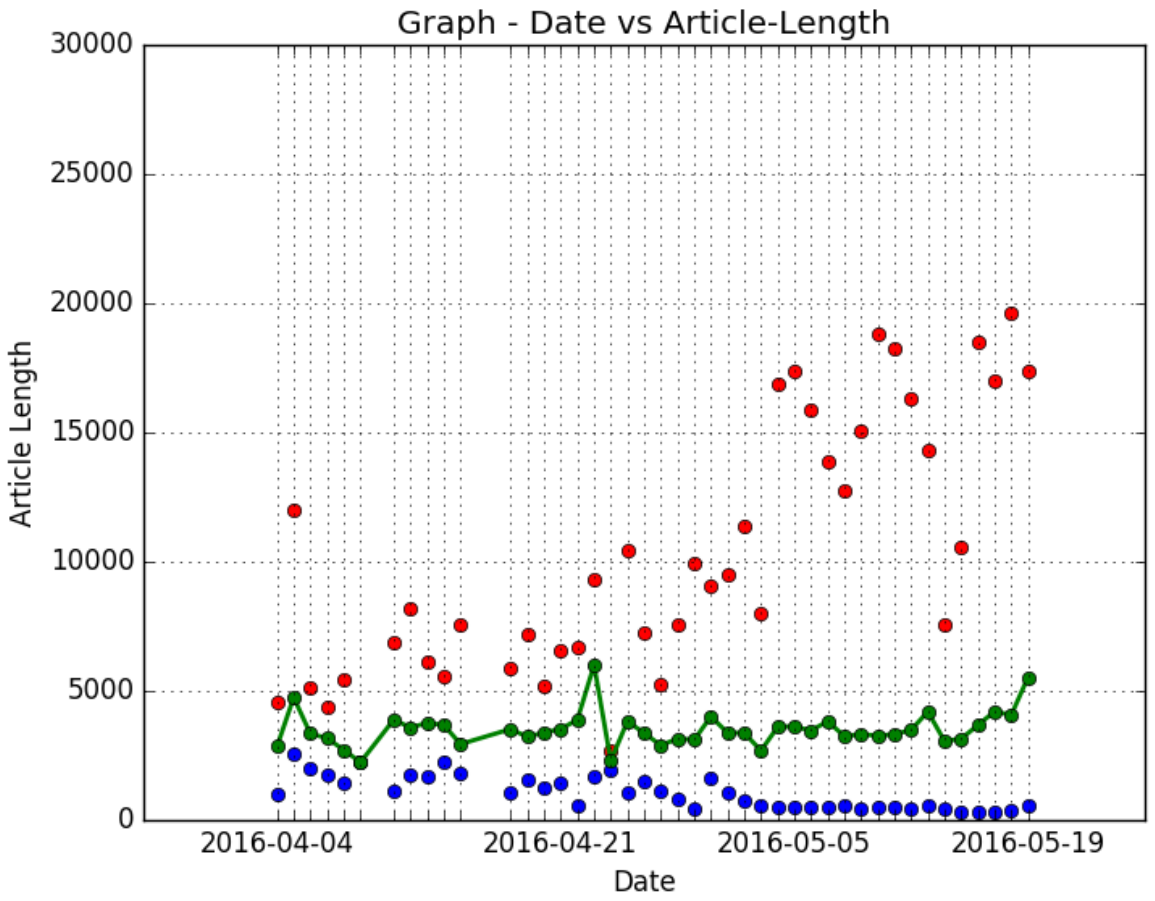


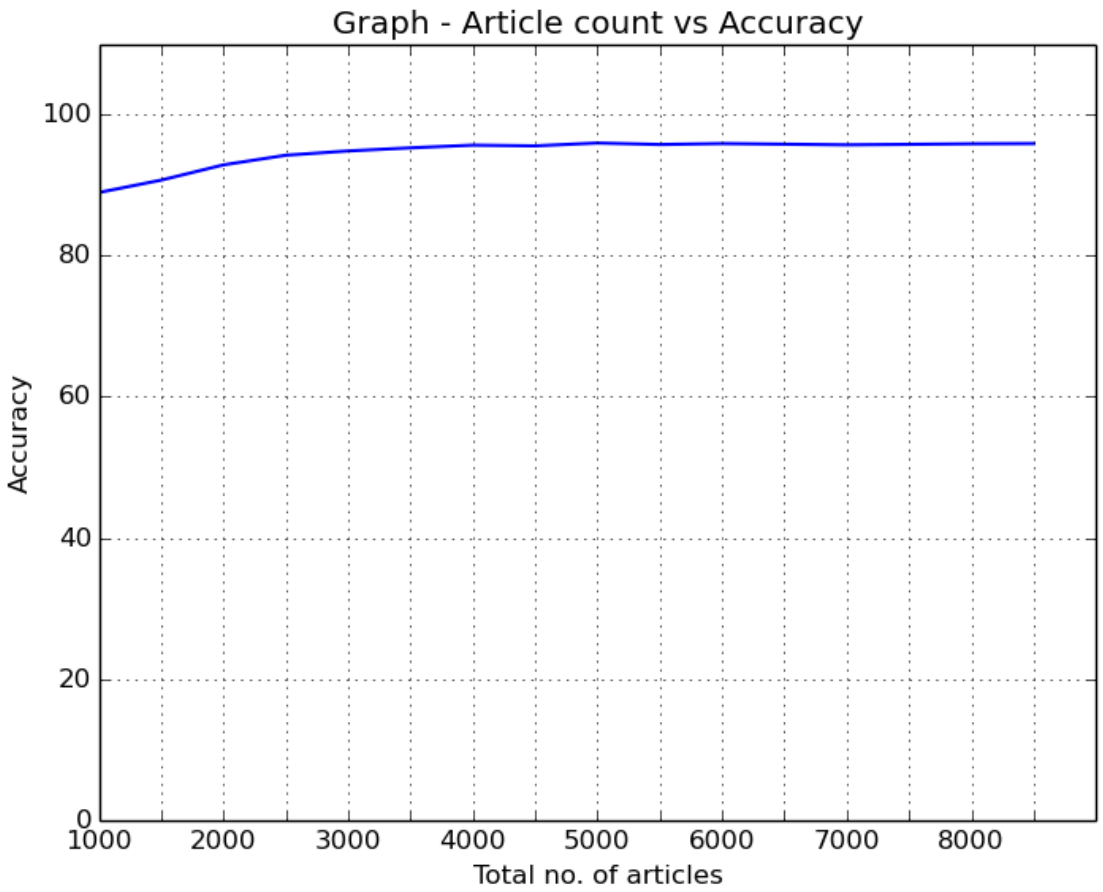
Analysis

- Random testing
  - Data set size : 5000+
  - Accuracy : 94.6
- Best case test
  - No. of fault classification : 265
  - Data set size : 9000+
  - Accuracy : 97.17

Best case Errors out of 9000+ articles

Analysis


Graph - Date vs Article-Length
30000
25000
20000
15000
10000
5000
0
Article Length
2016-04-04
2016-04-21
2016-05-05
2016-05-19
Date


Graph - Article count vs Accuracy
100
80
60
40
20
0
Accuracy
1000
2000
3000
4000
5000
6000
7000
8000
Total no. of articles

Proposed Enhancements

- Multi-category tagging for documents
- Dynamic categorization during fetching

Event detection

- Event detection is an unsupervised learning task.
- We used ***retrospective event detection*** which identifies previously unidentified events in chronological order.
- Events are those phrases which occur in a peaking frequency for a short period of time.

Event detection Procedure

Data selection and sampling

- Fetch data from database for a period

Transformation and reduction

- tokenize
- N-gramize

Unsupervised learning

- Compute weight for each phrases
- Compare with previous events

Analysis and verification

- Compare with online data manually
- visualization
- Compute hit and miss and false alarms of events. (Out of the program task)

Data transformation

- The data used for learning purpose must be cleansed before the task
- Main processing steps are
  - Tokenize the news data and remove stopwords.

    "Brain cancer detected in the Assam state

    { brain, cancer, detected, assam, state }
  - Create unigrams, bigrams, and trigrams and add it to the phrases list (learning vocabulary)

    { {brain, cancer, ... }

    {brain cancer, cancer detected, ... }

    {brain cancer detected, cancer detected assam, ... }

    }

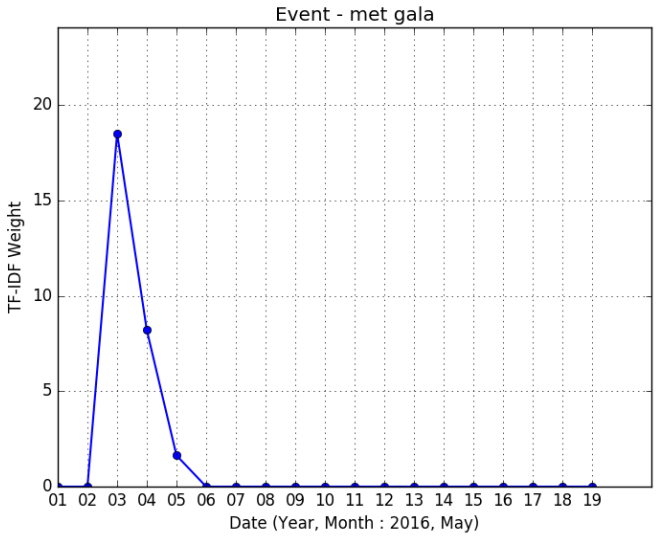
Event detection working

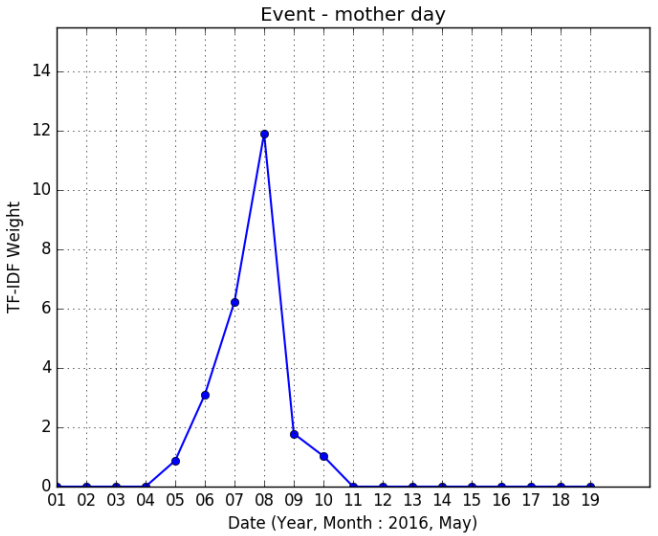
Weight of a term is calculated as,

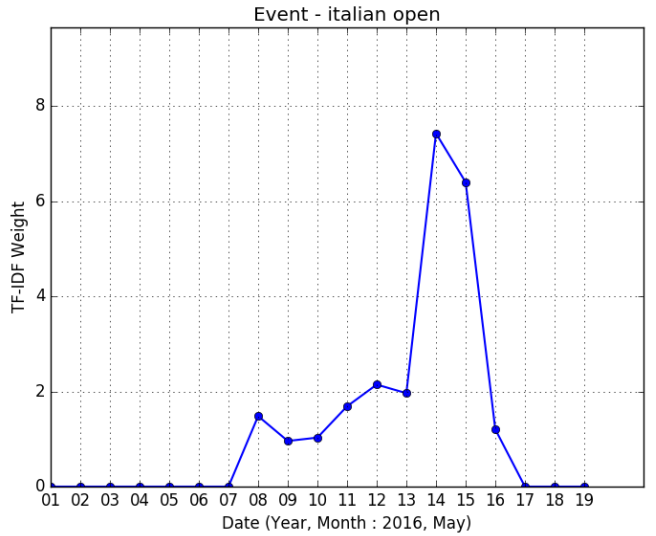
$$\boldsymbol{w(t,d) = \frac{\left(1+tf(t,d)\right) \times log(N/n_t)}{||\vec{d}||}}$$

- $w(t,d)$ *is the weight of term* t *in document* d
- $tf(t,d)$ *is the term frequency* (TF) $\quad tf(t,d) = a + (1-a) . \frac{f_{t,d}}{max_{\{t' \in d\}} f_{t',d}}$
- $\log(N/n_t)$ *is the Inverted Document Frequency*
- N *is the size of the training corpus*
- n_t *is the no. of documents containing term* t
- $||\vec{d}|| = \sqrt{\sum_t w(t,d)^2}$ is the 2-norm vector

Analysis





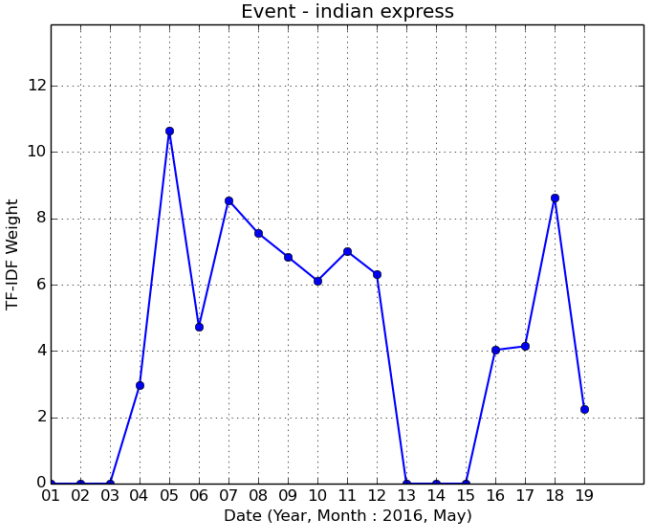


Some correctly detected events

- *Met gala*
- *ipl 2016*
- *Mother day*
- *Assembly polls*
- *Cannes review*
- *Rio Olympics*

Some noises

- *Film review*
- *Times India*
- *Jessica parker*
- *Indian express*
- *Captain America*
- *Net profit*



Proposed Enhancements

- Clustering and event merging.
- Real-time online reference, comparison, and verification.

References

- Learning approaches for Detecting and Tracking News Events: IEEE paper by Yiming Yang, Jaime Carbonell, Ralf Brown,Tom Pierce, Brian T. Archibald, Xin Liu
- Web references:
  - http://stevenloria.com/how-to-build-a-text-classification-system-with-python-and-textblob/
  - https://en.wikipedia.org/wiki/Tf%E2%80%93idf

Thank you….