Interim Report — CS 5604 Information Storage and Retrieval

CLA Team, Fall 2016

CLA Team:

Saurabh Chakravarty
Mahesh Narayanamurthi
Hyogi Sim
{saurabc,maheshnm,hyogi}@vt.edu

Project Advisor:

Prof. Edward A. Fox

September 20, 2016
Blacksburg, Virginia

Keywords: Text Classification, Information Retrieval, Information Storage

Interim Report — CS 5604 Information Storage and Retrieval

2016 Fall CLA Team

(ABSTRACT)

In this term project, we aim to design and develop a document classification technique that can complement the outcomes from the previous semesters. Particularly, our design is focused on improving the following aspects, in classifying the huge tweet collections into predefined categories. First, the result of the classification should be reliable so they can be practically utilized in decision makings. To this end, we plan to achieve higher accuracy, by exploring and adopting the state-of-the-art algorithms, than the previous classification techniques provide. In addition, an intelligent spam filtering technique will also be studied and developed. Second, the classification system should perform in a scalable manner. The amount of data that data analysis softwares need to ingest is increasing unprecedentedly, and, therefore, it is crucial for such softwares to provide deterministic and scalable performance for effectively supporting decision making processes. To achieve such scalability with respect to the increasing amount of the data, our classification software will attentively orchestrate the I/O behaviors of individual worker threads by considering the underlying system framework.

This interim report clarifies the problem that we address along with our project goals, and summarizes related theories and practical techniques, We then lastly provide our concrete execution plans.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we describe the classification problem that we will address in our term project (§ 1.1), we state the assumptions with which we will be working and give a brief overview of the strategy that we will follow, details of which can be found in (§ 3.1).

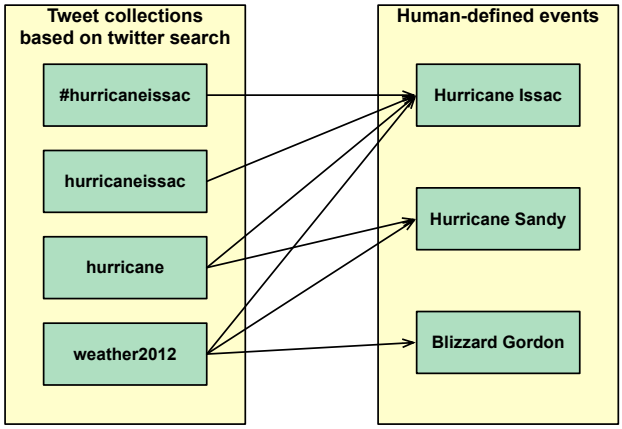

Figure 1.1: Problem statement.

1.1 Requirements

The goal of the classification team is as follows:

> Given a tweet collection and associated content from webpages linked in the tweets, and a set of events in the real world, we are to build a classifier that can classify the tweet and the webpages to a given event class.

Figure 1.1 explains the problem pictorially. Essentially, we have a collection of tweets that have been retrieved based on keyword/tag search performed using the Twitter API. Additionally we have the content from webpages linked in the tweets. We will call this webpage and tweet collection together as documents in the remainder of this report, with exceptions to this convention noted explicitly. These are shown on the box on the left in Figure 1.1. The human defined events or the real life events as stated in the goal are shown in the box on the right. The relationship between the collection of tweets and the events is many-to-many.

For instance, the tweet collection `weather2012` can have tweets related to the hurricanes "Sandy" and "Issac" as they occured in the same year and tweets from this collection can map to either "Hurricane Sandy" or "Hurricane Issac" event on the right. Likewise, for a given event, there can be several tweets that are associated with the event. We would also like to note that for a given event, just as there can be several tweets associated with the event, there can be multiple webpages that can be associated with the event. In a similar fashion, there could be websites that are comparing past with current events and could be classified in either event category.

For the task of classification, we make the following assumptions about the collection of tweets and webpages:

- Tweets have been extracted and are available as CSV files and some "basic" SPAM check has been done by tweet collection management team or by the teams from previous offerings of this course.
- Webpage content will be extracted and be stripped of all tags and unrelated content by the website collection management team or by the teams from previous offerings of this course.
- We will provide the Solr team with tweet id/webpage id and tags to classify the tweets and associated webpages.

More assumptions will be made as we make further progress on the project. We will now describe the approach that we will resort to from a very high level:

1. We start the process with a set of clean tweets and webpage content.
2. We perform some pre-processing to either load the data or extract only relevant parts of it.
3. Next we annotate the collection as either being relevant to a particular category or not.
4. We generate statistics related to the data obtained in step 2.
5. The statistics are used to select features.
6. A training model is selected and using the annotated data and the set of features, its trained on.

7. This trained model is then used in classifying a larger collection of data.

Additionally we may resort to other techniques such as bootstrap strategy to handle the larger datasets and these will become clearer in the weeks to come.

Chapter 2

Related Work

We went through some papers that had done an extensive literature survey in this field. We plan to go through the sections on comparative analysis of the different feature selection methods described in [1]. We also plan to go through the various techniques on different document representation methods in [1].

Chapter 3

Design

3.1 Design of the Classification System

We will use the following high level approach as shown in Figure 3.1.

- For each human defined event, we will manually identify what tweet collections can be used that might contain relevant tweets.
- We assume that the document collections have been cleansed of noise, SPAM and other artifacts occurring in tweets. Some of the pre-processing of this kind has been done by the class of Spring 2016 and saved in HBase collections. We will use these refined collections for our work as much as possible.
- We might perform some additional noise and SPAM cleansing based on what we observe from going through the noise/spam related artifacts from the refined document collections.
- We will create training data out of the document collections by going though the tweets manually and classifying them as relevant/non-relevant. To save time, we will use the results obtained by the previous class as much as possible owing to their availability.
- Once we have the labelled training data, we will do some statistical analysis of the tweets based on the following techniques.
  - Identify the top-k relevant words that correlate highly with the actual event. Sorted TF-IDF scores.
  - Top-k association rules.
  - Frequent pattern mining.

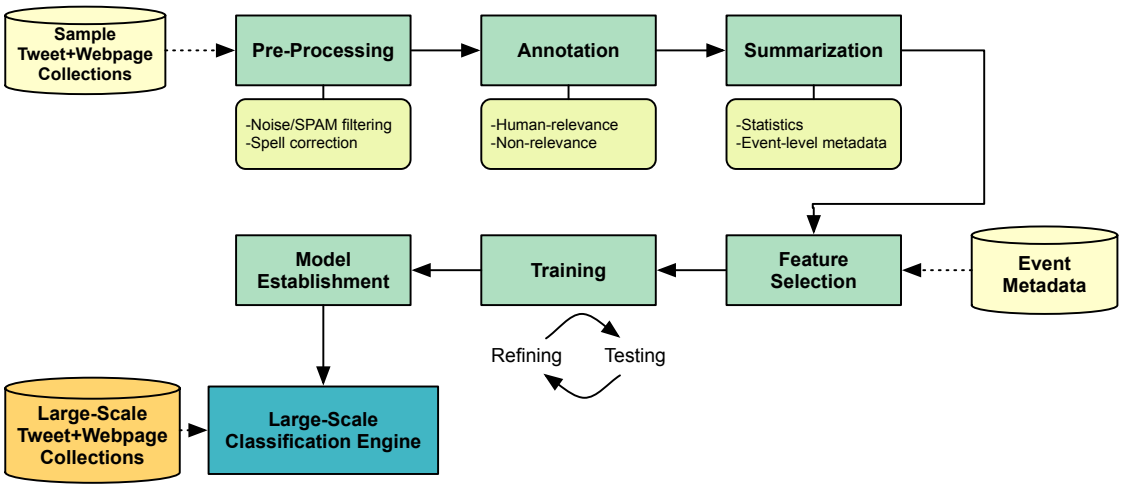

Figure 3.1: Classification pipeline.

Field	Description
Collection ID	The tweet collection ID
Collection Filter	Description
Event Name	The name of the event
Event Tags	[**H**] Related tags
Event Keywords	[**H**] Related keywords
Important Words	Important words inferred by the classification processing
Important Phrases	Important phrases inferred by the classification processing
TF-IDF best words	Best words suggested by TF-IDF
Mutual Information	Expected MI computed by the classification processing

Table 3.1: Classification metadata.

- The results of the previous step will be saved as part of the event metadata for each human defined event. This metadata will be used to compute features for a given tweet, during trainings, test and classification time. This metadata will be saved in a HBase collection.
- Once we have tuned the parameters of the classifier, we will save the generated model in HDFS. We will deploy this model in the cluster and will perform classification on the large document collections. Please refer to Table 3.1.
- The classified tweets will be saved back in HBase with the relevant event tags.

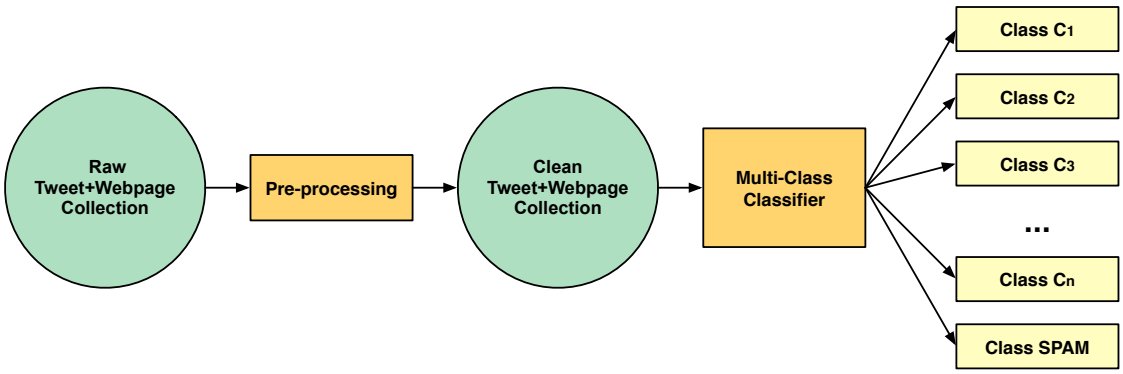

Figure 3.2: Training data approach.

Chapter 4

Execution Plan

We plan to study the related techniques in the literature first. Then, based on the progress, we plan to implement and evaluate our classification algorithms.

Chapter 5

Conclusion

In this report, we have clarified the problem, goal, and execution plan of the document classification team.

To summarize, we will classify the tweet and web-page collection based on the association rule-based approach. In addition to the algorithmic aspect, we will also try to reduce the classification time by exploiting the benefit of in-memory data analytics framework, e.g., Apache Spark.

Bibliography

[1] D. A. Pereira, E. E. B. da Silva, and A. A. A. Esmin, "Disambiguating publication venue titles using association rules," in *Digital Libraries (JCDL), 2014 IEEE/ACM Joint Conference on*, pp. 77–86, Sept 2014.