News Event Detection

By Q Liu(Huanqing), Andrew Ciambrone, Alex Cummins, Beakal Haile

Client: Mohamed Magdy Faculty
Advisor: Dr. Edward Fox Course: CS
4624
Date: 04/27/2015

Table of Contents

Executive Summary

The purpose of this system is to be able to automatically detect when a new world event has occurred using RSS feeds of popular world news websites. This will aid social scientists in their analysis of world trends. This system is being designed specifically for integration with the the IDEAL (Integrated Digital Event Archiving and Library - http://www.eventsarchive.org/) project. The system is a collection of Python scripts each of which fulfils an objective outlined in greater detail in Section 3 System Architecture.

It is pertinent that we begin work on this project in a timely manner because under the current system there is much wasted labor and the potential failure to detect an event. If we do not complete this project now there's the potential for missing the detecting of an event which will lead to a failure to archive said event resulting in permanent loss of that data. Additionally our client will have to waste excess time searching twitter and website feeds in order to manually detect events. If we didn't do this our client would have to do this manually every day or spend the time implementing it himself preventing him from performing other duties that could be more critical.

1 Application Overview

1.1 Objectives

1.1.1 Beneficiaries

There is the potential for many parties to benefit from the success of this project. The IDEAL team will benefit directly from this project because it would be a noticeable augmentation to their current system. Foreign policy makers could benefit from a concise summary of world events, similar to the report the President of the United States receives each morning. More broadly anyone who would like to keep up to date with current events can benefit from this project. The primary objective of this project comes in the form of saved time. Instead of looking through several news websites to find out new events the user can use this project to find the most significant news of the day.

1.1.2 Possible Alternatives

We are confident that automating new event detection will be significantly advantageous to the IDEAL project in order to merit our efforts in developing the system. While the alternative method of manual event extraction exists it is too time consuming to be sustainable in the long run.

1.2 Business Process

1.2.1 Existing Systems

Currently, our client is manually reading through Twitter feeds and various news websites to detect a new event. This process is very tiring and cumbersome.

1.2.2 Future Business Processes

The finished project will consist of a working pipeline of several technologies that when started will aggregate news events from several popular news site's RSS feeds as well as Twitter feeds. These news events will be stored in a MySQL database to be processed by an algorithm that detects if the news event is concerning an important new event, using natural language processing libraries. The output of the pipeline will be prioritized for correctness of classification as a new event over quantity of new events to reduce false positives.

1.3 User Roles and Responsibilities

There will be three primary users of the the system. The person who installs the software and the person who maintains the software. The person who installs the software will be responsible for following the developers guide to correctly set the system. The person who maintains the system will need to check a log for any errors that may have occurred during the normal usage of the system.

1.4 Interactions with Other Systems

The system will be integrated with the current IDEAL system. It is the beginning of the pipeline for archiving information on world events. The output will need to formated in a manner that is compatible with the later stages of the system.

2 Functional Requirements

2.1 State of Functionality

The finished project will consist of a working pipeline of several technologies that when ran will collect news articles from several popular new site's RSS feeds as well as Twitter feeds. Next, these articles will be stored in a MySQL database to be processed by an algorithm that detects if the collection of articles is reporting any important new events. The output of the pipeline will be prioritized for correctness by using strength analysis techniques and through classification to prevent or reduce the number of false positives.

Part A: Event Detection

1) Identify a list of websites to be used in determining the occurrence of an event.
2) Establish a stream of input from the identified websites.
3) Detect a new event from the given stream of stories based on some threshold.
4) Develop a ranking system to convey the certainty that the detected event has occurred.

Part B: Event Categorization

1) After an event has been detected determine what category it falls into.
2) Prepare training data
3) Train classifiers

Part C: Documentation

1) User manual
2) Developer manual
3) Installation instructions
4) Procedure
5) Performance report

2.2 Administration/Customization of the Application

Due to the desire for the broad usage of this application it will be simple to implement for a technical user. The program will come with a default list of sources from which to collect data but the list of inputs will be modifiable. The output of what events were detected will be modifiable based on a threshold of the likelihood that an event has occurred. In this way the end user will be able to modify the system's threshold for at the risk of misreporting. The system should require little maintenance.

2.3 Reporting

Bi-weekly progress reports will be delivered to the client. In these reports the team will debrief the client on what advancements have been made to each part of the system. If the team is blocked on something they will be sure to contact the client as soon as possible so the client is able to respond to the issue in a timely manner.

2.4 Scope

The scope of this problem is very open ended; however, as a minimum requirement our client would like the project to be able to collect and detect new events from English news websites that have an RSS feed. Furthermore, the new events should be categorized appropriately.
As time permits the project can be extended to be able to collect news from websites that do not have RSS feeds or even social media such as Twitter and Facebook.

2.5 Performance

Creating a high performance system will be quite difficult to implement. The goal of this project is to develop a fully functional prototype and then tune for performance as time permits. In order to achieve the performance goals the development team will train the system against classifiers. The classifiers will be created and tuned as the development proceeds.

2.6 Usability

A priority expressed by the client was a need for correctly classified output over quantity of classified output. To ensure that this priority is met the classification algorithm will be built with a extremely low false positive rate.

3 System Architecture

3.1 Architectural Design

The flow of information through this system is linear, making the design relatively simple. The information flow begins with the collection of newsworthy data and then travels through a series of analytical stages outputting a list of new topics. In these analytical stages the textual similarity among the documents is first calculated. Then heuristics are applied to determine a confidence level of event occurrence. If the confidence level is above the desired threshold than we say an event has occurred. In order to determine if it is a new event we simply compare it to the results from the previous timeframes.

3.2 Decomposition

The process of decomposition is split into several components outlined below.

3.2.1 Collection of Articles

This stage in the process begins with the identification of sources from which to establish a stream of data. Due to time constraints in this project selecting a few sources with relatively clean data makes the most sense. Otherwise an inordinate amount of time might be spent cleaning the data as opposed to doing the actual analysis. Once the sources are identified, web scrapers to collect the information must be developed. It is advantageous to store as much historical information about events as possible. As such it is impractical to store all of that information in memory so a data storage system must be developed. The data stream's format lends itself to being stored in a relational database. After the sources have been identified, the scrapers are pulling information, and the database is populated, it is possible to move to the next module.

3.2.2 Document Similarity Matrix

In this module the textual similarity between documents is calculated. In the field of natural language processing it is common to represent a document as a multidimensional vector. Each dimension represents a word and the magnitude is the number of times the word appears in the document. In this project our motivation for vectorizing documents is then taking the cosine between the two vectors this results in a score from one to zero on how similar the documents are. This is a very computationally expensive process so we will attempt to speed it up in by reducing the number of dimensions in the vector space. First we remove common words that fail to convey meaning; these are often referred to as stop words. Second we stem the words by removing prefixes and suffixes. Stemming the words has the added advantage of producing better results when calculating the cosine similarity. The cosine similarity values will then be used in the creation of clusters.

3.2.3 Cluster Creation

This module will consume the document similarity matrix produced from the previous model and generate a list clusters that represent potential topics. The difficulties faced in creating the clusters include deciding what threshold of cosine similarity to use and optimizing for speed. Deciding what threshold to use to create the cluster will have to be tested and refined in a few iterations. In the most obvious implementation this algorithm will run in $O(n^2)$ time where n is the number of articles. This means the algorithm will not scale well. However it will be doing a lot of duplicate computations so a lot of the work done in this module will be in reducing those.

3.2.4 Cluster Strength Analysis

This is the third stage in the pipeline and this stage is to determine the strength of the cluster. The purpose of this module is to determine how sure it is that an event has occurred. We are going to perform this task by counting the number of unique publishers and dates from a given list of articles. The number of articles and number of dates will go into a formula and output a score between 0 and 1 with 1 being we are 100% sure an event has occurred. For a more detailed explanation refer to section
Several stages of refinement will occur in this module.

3.2.5 Classification

In this module we are classifying the type of event that might be going on. The client asked that we classify events into categories such as plane crashes, war conflict, and natural disasters. We will be given the title and description of a number of documents and asked to use a method of document classification, such as using Naive Bayes Classifiers or other classifiers available in the weka package. This will require other classifiers, to be 'trained' to put certain documents into a category. Classification is important not only because it will be a useful tool for the client but it will also help in the cluster strength analysis. We can assume if a number of documents all show a certain type of event is happening then we can be more certain that those documents are suggesting an event is occurring.

3.2.6 Testing

Testing this system is rather difficult due to the ambiguity surrounding what constitutes an event. Even discounting that ambiguity it is not sufficient to assemble a list of events that have occurred by examining the data we have collected. Our data may not be representative enough to contain all pertinent world events. Taking these shortcomings into account we will attempt to leverage a domain expert in the area of foreign policy so that we have. This individual will provide us with a list of events and the times at which they occurred. Then it will be possible to refine the algorithm in an attempt to obtain similar results from the data.

4 Implementation Plan

4.1 Implementation Overview

Since this the problem this project solves is rather difficult it uses several different technologies at its disposal. The source code for the project will be stored on the code repository service bitbucket.

4.1.1 Technologies

Like previously stated to store the articles being collected we use an sql database which is hosted on an Amazon Web Service server. The program itself will be run on a large cluster machine itself due to the fact that many of the algorithms used by the program are CPU intensive.

4.1.2 Programming Languages

Since this project is only a part of a larger project the client has asked us to use the programming language Python. This presents us no real constraints considering python has many libraries for data analytics, mySQL and natural language processing.

4.2 Implementation Details

The following is an detailed overview of how we are collecting news, organizing the news, and determining if the news is a new event.

4.2.1 Collection of Articles

Articles will be collected from the following news sources Al Jazeera, BBC, Xinhuanet, New York Times, Wall Street Journal, Financial Times, Reuters, and CNN. Scraping of these sources will be accomplished using Pythons web scraping API. The Beautiful Soup library for Python will be used to parse the HTML that was gathered. Beautiful Soup is well known and supported. The HTML will be parsed for the following items title, description, link, published date, and language. Additionally the date of collection will be recorded and the publisher will be deciphered from the url. All of these item will be added to a mySQL database hosted on Amazon Web Services. The mySQL database will consist of a single table that contains all of the information about each article.

4.2.2 Document Similarity Matrix

Python's has a very well known library for natural language processing aptly named the natural language toolkit (nltk). To create the document similarity matrix all articles collected from the past three days will be queried from the mySQL database. We consider an article in this case to consist of the title and description components of the RSS item. Each article will have all of the stop words removed from it using nltk's corpus for english stop words. Then the term frequency–inverse document frequency a statistic meant to intended to reflect how important a word is to a document in a collection will be calculated using nltk. Each of these values will be used to create a vector representation of each document. Finally all the documents vectors will be compared using nltk's cosine distance function. These values will be saved into the document similarity matrix.

4.2.3 Threshold Filtering

In order to start grouping the rows and the related column in the matrix we need to create groups of documents that are about the same news event by cutting them off at a certain threshold. For example, if our matrix looked like this:

	Document 1	Document 2	Document 3
Document 1	1	0.65	0.33
Document 2	0.65	1	0.67
Document 3	0.33	0.67	1

Figure 1: Example of similarity matrix that is generated where the ith row and jth column shows the cosine similarity between documents i and j.

With a threshold of .5 document 1 would be grouped with document 3, document 2 would not be grouped with anyone and document 3 would be grouped with document 1. Groups of similar documents will be referred as clusters. This would mean document 1 and document 3 are similar. As you can see this is a inverse matrix and calculations can be sped up knowing this fact.

4.2.4 Difficulties in Cluster Strength Analysis

Traditional clusters produced using cosine similarity as we are doing is simply not effective enough at determining whether or not 2 documents are actually talking about the same event. This is due to the fact that natural language processing is still in its infancy and that language is incredibly complex. One main reason is that there is a lot of ambiguity in language such as syntactic ambiguity and words with more than 1 meaning or that meaning. Another main reason is that meaning is context sensitive. For example that sentence "Let's go eat" true meaning depends on time of day. In news events titles of articles and the articles themselves may assume readers already have knowledge of the time of day from a picture we cannot analyze or that readers already have knowledge of social context, location, and prior news events.

In order to help analyze the strength of the clusters(groups of similar documents) we will leverage the additional data collected about each article, namely the date collected and the publisher. We will count the number of distinct values for each of these field and generate certainty of event occurrence value between zero and one. Where one represents absolute certainty that an event has occurred.

4.2.5 Classification

In terms of the classification portion of the system there is going to be several steps of refinement. In the initial stage classification is going to be rudimentary and only involve checking the titles and their descriptions and seeing if they contain any of the key words for the given predefined categories. For example if there were a plane crash category such as malaysia airline crashes then we would check the article title for "Plane", "Crash", and "Collision". While this may seem simple it will give us a basis for testing the other sections of the system so that they be better refined. The next step in the refinement process would be to implement a whole new method for document classification. If there is time our client has proposed we use K means classification algorithm. Using the classifiers found in the nltk's weka package this new method would be trained using positive and negative examples. Training the classifier is where most of the work is currently being done since numerous examples are needed for the classifier to be accurate. Which is why a basic implementation of the classifier was first established so that the rest of the team was not held back.

4.2.6 Testing

Testing is a very critical part of our development cycle since many of the parts need to be refined so that we can reduce the number of false positives. To test our

system we plan to run our system on given test data and compare the results of our system to the results given by an expert in foreign affairs who is also given the same test
data. From this we will tweak our thresholds to reduce the number of false positives and create a more reliable system.

4.3 Gantt Chart

Task	Percentage Completed	Status	Day Started	Day to Be Complete	Responsible Party
Set up amazon ec2 instance for running scraper	100%	Completed	2/4/2015	2/5/2015	Huanqing Liu
Set up Bitbucket and everyone’s ssh keys	100%	Completed	2/4/2015	2/8/2015	Huanqing Liu
Create Scrapers	100%	Completed	2/4/2015	3/4/2015	Alex Cummins
Document Similarity Matrix	100%	Completed	2/4/2015	3/4/2015	Alex Cummins
Strength Analysis	100%	Completed	N/A	3/19/2015	Beakal Haile
Classify	100%	Completed	2/23/2015	3/19/2015	Andrew Ciambrone

1st Demo	100%	Completed	N/A	4/2/2015	N/A
Assign work based on 1st Demo Feedback	0%	On Schedule	4/2/2015	4/2/2015	Everyone

5 Prototype Overview

Over the last few weeks we have been working closely with our client to create a working prototype. We are proud to say that a crude but functional prototype of our system has been implemented and in this section we will go into depth about how we implemented our prototype.

5.1 Working prototype

The most up to date prototype code can be viewed by going to https://bitbucket.org and logging in with the following credentials that have been provided with view only access.

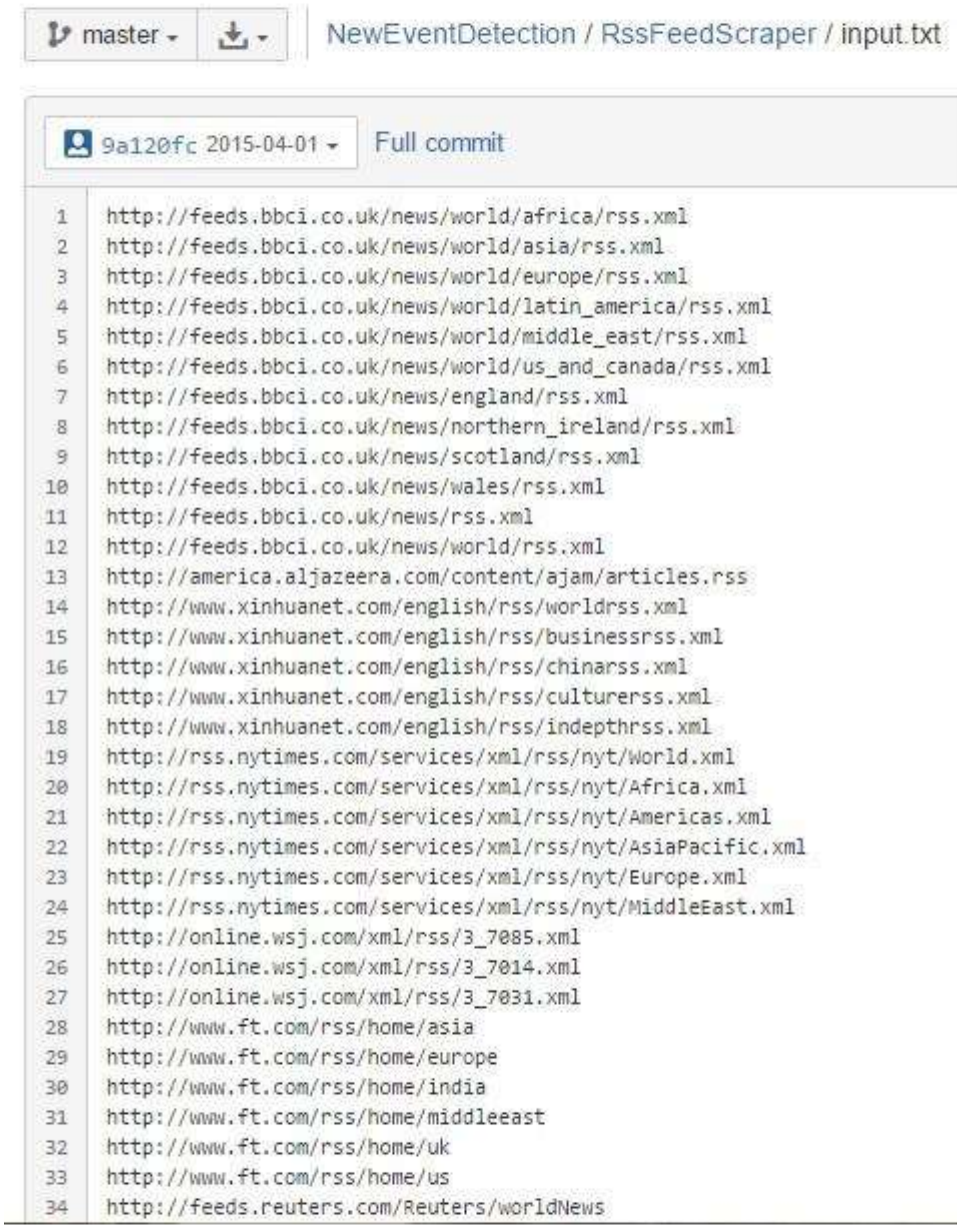
Username: hypertextclient
Password: ionlywantnewnews

Then proceed to visit: https://bitbucket.org/quinnliu/neweventdetection

The following is a screenshot walkthrough of the repositories code used to collect news articles, generate the document similarity matrix, and generate the final list of new news events since there are currently no UI elements.

5.1.1 News article collection

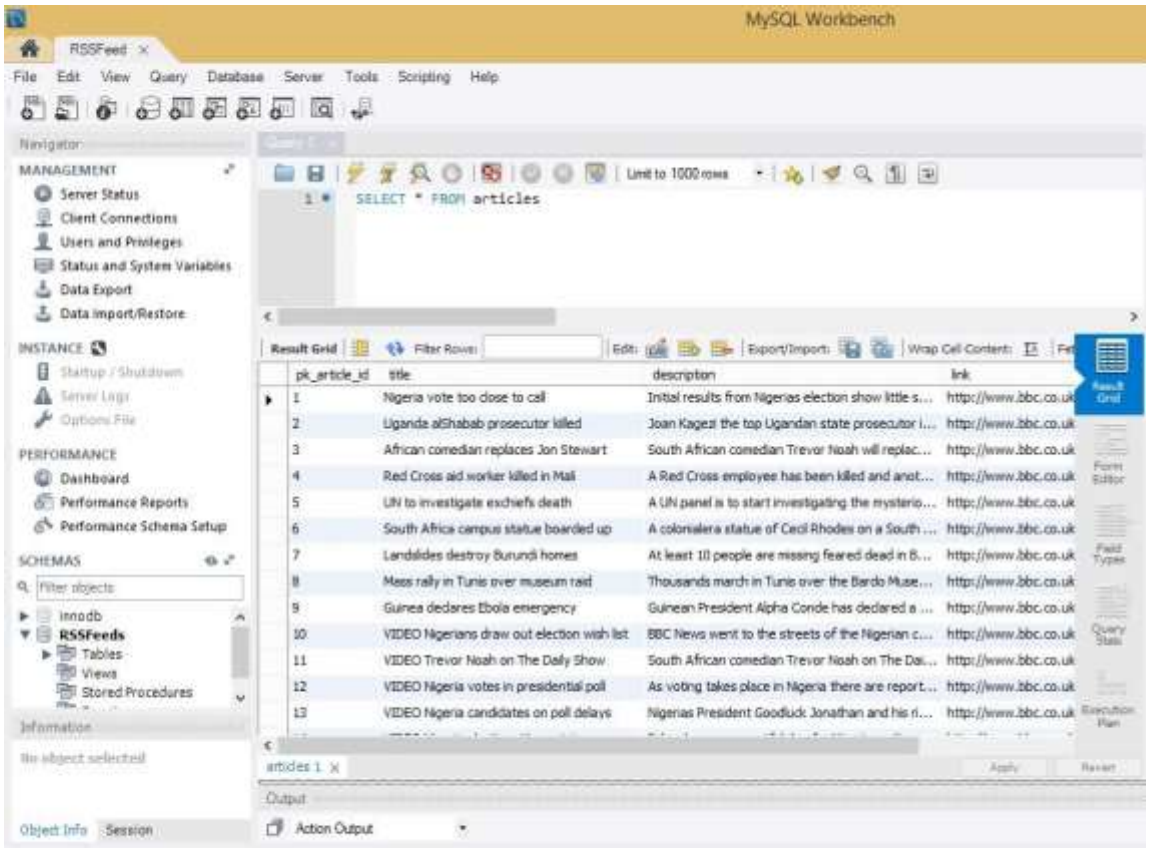
Our code is hosted in a remote machine running a script every 6 hours to collect new news events. In Figure 2 we see the list of RSS feeds of popular news websites we are currently gathering news information from.



```
http://feeds.bbci.co.uk/news/world/africa/rss.xml
http://feeds.bbci.co.uk/news/world/asia/rss.xml
http://feeds.bbci.co.uk/news/world/europe/rss.xml
http://feeds.bbci.co.uk/news/world/latin_america/rss.xml
http://feeds.bbci.co.uk/news/world/middle_east/rss.xml
http://feeds.bbci.co.uk/news/world/us_and_canada/rss.xml
http://feeds.bbci.co.uk/news/england/rss.xml
http://feeds.bbci.co.uk/news/northern_ireland/rss.xml
http://feeds.bbci.co.uk/news/scotland/rss.xml
http://feeds.bbci.co.uk/news/wales/rss.xml
http://feeds.bbci.co.uk/news/rss.xml
http://feeds.bbci.co.uk/news/world/rss.xml
http://america.aljazeera.com/content/ajam/articles.rss
http://www.xinhuanet.com/english/rss/worldrss.xml
http://www.xinhuanet.com/english/rss/businessrss.xml
http://www.xinhuanet.com/english/rss/chinarss.xml
http://www.xinhuanet.com/english/rss/culturerss.xml
http://www.xinhuanet.com/english/rss/indepthrss.xml
http://rss.nytimes.com/services/xml/rss/nyt/World.xml
http://rss.nytimes.com/services/xml/rss/nyt/Africa.xml
http://rss.nytimes.com/services/xml/rss/nyt/Americas.xml
http://rss.nytimes.com/services/xml/rss/nyt/AsiaPacific.xml
http://rss.nytimes.com/services/xml/rss/nyt/Europe.xml
http://rss.nytimes.com/services/xml/rss/nyt/MiddleEast.xml
http://online.wsj.com/xml/rss/3_7085.xml
http://online.wsj.com/xml/rss/3_7014.xml
http://online.wsj.com/xml/rss/3_7031.xml
http://www.ft.com/rss/home/asia
http://www.ft.com/rss/home/europe
http://www.ft.com/rss/home/india
http://www.ft.com/rss/home/middleeast
http://www.ft.com/rss/home/uk
http://www.ft.com/rss/home/us
http://feeds.reuters.com/Reuters/worldNews
```

Figure 2: Screenshot of first 34 RSS feeds we are using as input.

To collect the actual news articles we are using the Pydev eclipse development environment with code for taking iterating through RSS feeds in input.txt to save in MySQL database. You can see the MySQL database with all of the articles displayed using the SQL command “SELECT * FROM articles” in Figure 3.



```
SELECT * FROM articles
```

pk_article_id	title	description	link
1	Nigeria vote too close to call	Initial results from Nigerias election show little s...	http://www.bbc.co.uk
2	Uganda alShabab prosecutor killed	Joan Kagezi the top Ugandan state prosecutor i...	http://www.bbc.co.uk
3	African comedian replaces Jon Stewart	South African comedian Trevor Noah will replac...	http://www.bbc.co.uk
4	Red Cross aid worker killed in Mali	A Red Cross employee has been killed and anot...	http://www.bbc.co.uk
5	UN to investigate exchiefs death	A UN panel is to start investigating the mysterio...	http://www.bbc.co.uk
6	South Africa campus statue boarded up	A colonialera statue of Cecil Rhodes on a South ...	http://www.bbc.co.uk
7	Landslides destroy Burundi homes	At least 10 people are missing feared dead in B...	http://www.bbc.co.uk
8	Mass rally in Tunis over museum raid	Thousands march in Tunis over the Bardo Muse...	http://www.bbc.co.uk
9	Guinea declares Ebola emergency	Guinean President Alpha Conde has declared a ...	http://www.bbc.co.uk
10	VIDEO Nigerians draw out election wish list	BBC News went to the streets of the Nigerian c...	http://www.bbc.co.uk
11	VIDEO Trevor Noah on The Daily Show	South African comedian Trevor Noah on The Dai...	http://www.bbc.co.uk
12	VIDEO Nigeria votes in presidential poll	As voting takes place in Nigeria there are report...	http://www.bbc.co.uk
13	VIDEO Nigeria candidates on poll delays	Nigerias President Goodluck Jonathan and his ri...	http://www.bbc.co.uk

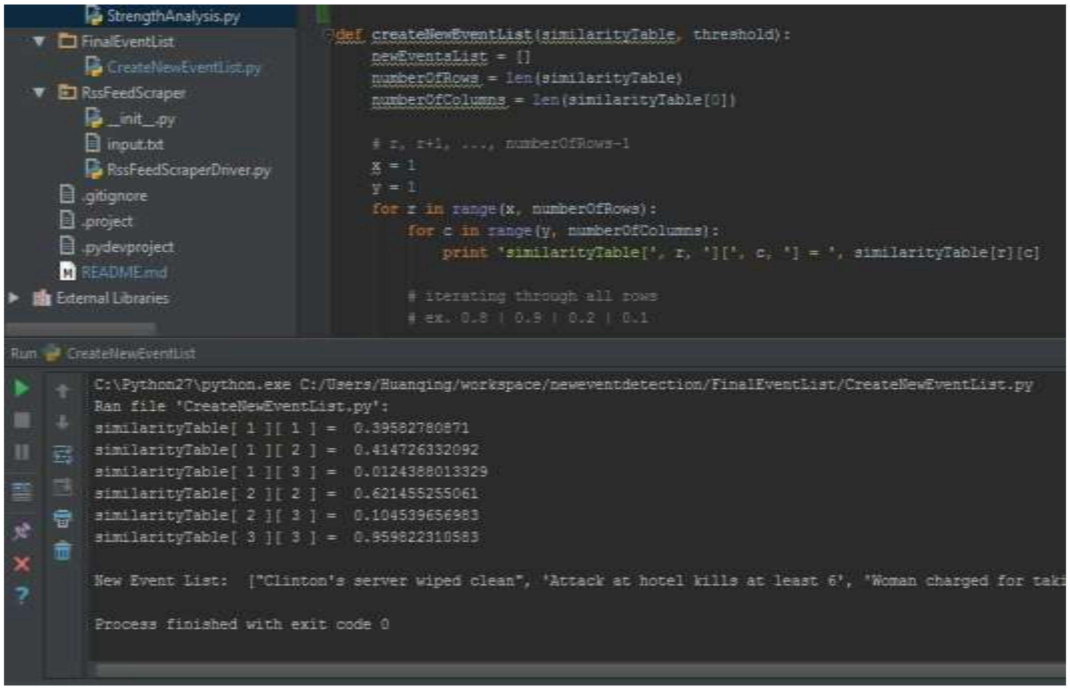
Figure 3: Screenshot of MySQL workbench displaying collection of news articles.

5.1.2 Document Similarity Matrix

To create the document similarity matrix we use nltk's cosine distance for calculating initial similarity between documents. The code for creating the similarity matrix is in the folder Cluster/cosine_similarity.py by following the instructions for viewing all code on bitbucket in the beginning of section 5. First we get the documents form the MySQL database and remove all stop words such as "the", "am", and "I". Then the nltk's cosine distance is used.

Next, there is additional checking of similarity between documents by checking date published and number of news articles on the same topic. This is called the strength analysis algorithm and the code is in the file "Cluster/StrengthAnalysis.py". First the algorithm takes into account how many articles we have in a cluster. This component is given 40% of the total weight when determining our level of certainty about an event occurring. Secondly, we give 40% for the total weight for the number of distinct publishers present in the cluster. Basically the more publishers we the more we are confident we are about an event occurring. Finally, the algorithm count in how many distinct days the articles are published. This component is given 20% weight.

After all the documents have been compared a matrix representing similarity between all documents will be generated. We iterate through the matrix and find multiple documents with high similarity to ensure we do not generate a final list of new news events with duplicates. In Figure 4 a final list of new news events will be displayed as a python array.



```
def createNewEventList(similarityTable, threshold):
    newEventsList = []
    numberOfRows = len(similarityTable)
    numberOfColumns = len(similarityTable[0])

    # r, r+1, ..., numberOfRows-1
    x = 1
    y = 1
    for r in range(x, numberOfRows):
        for c in range(y, numberOfColumns):
            print 'similarityTable[', r, '][', c, '] = ', similarityTable[r][c]

        # iterating through all rows
        # ex. 0.8 | 0.9 | 0.2 | 0.1
```

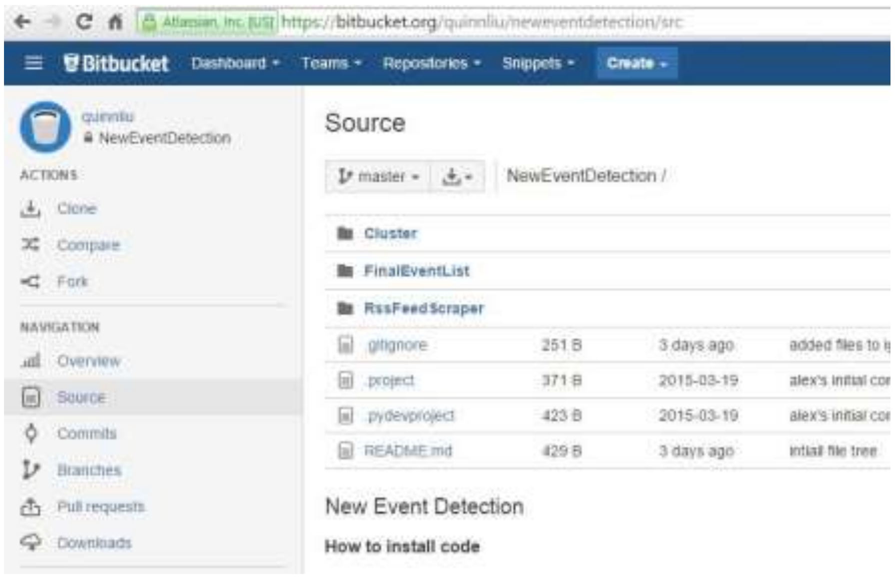
```
C:\Python27\python.exe C:/Users/Huanqing/workspace/neweventdetection/FinalEventList/CreateNewEventList.py
Ran file 'CreateNewEventList.py':
similarityTable[ 1 ][ 1 ] =  0.39582780871
similarityTable[ 1 ][ 2 ] =  0.414726332092
similarityTable[ 1 ][ 3 ] =  0.0124388013329
similarityTable[ 2 ][ 2 ] =  0.621455255061
similarityTable[ 2 ][ 3 ] =  0.104539656983
similarityTable[ 3 ][ 3 ] =  0.959822310583

New Event List:  ["Clinton's server wiped clean", 'Attack at hotel kills at least 6', 'Woman charged for taki

Process finished with exit code 0
```

Figure 4: Screenshot of oversimplified similarity matrix and final list of news events.

5.2 Changes and Extensions Planned

Before coding the project our team initially started with a github repository to store all code for our project. However we moved our code from github to bitbucket after we realized we needed a key to access the amazon database instance within our code. Bitbucket is a service that provides free private repositories for developer teams less than 5 people.

The first extension to our current pipeline is adding a classification of news event feature using simple keyword classification. After finishing our initial pipeline we would proceed to test our pipeline with a python unit tests of each method and also full regression tests that use the entire pipeline. Additionally if time permits a simple website will be hosted using heroku to display the list of news events outputted in Figure 5 in a readable format.



Cluster			
FinalEventList			
RssFeedScraper			
.gitignore	251 B	3 days ago	added files to i
.project	371 B	2015-03-19	alex's initial co
.pydevproject	423 B	2015-03-19	alex's initial co
README.md	429 B	3 days ago	intial file tree

New Event Detection

How to install code

Figure 5: Screenshot of our bitbucket repository.

6 Refinements 1

During our latest meeting with our client we presented our prototype of our project to our client. During our prototyping session we received feedback on both what was working well and what needs improvement. The main areas that need improvement include accuracy of our output, making our implementation reusable by making it table driven, and adjusting our weight system. In addition our client suggested a couple of features we should add. Furthermore our client has recommended that we add more news sources as well as refine what we currently have. In the following sections we will discuss more in depth about the improvement needed and how we plan on accomplishing our goal.

6.1 Improve Accuracy

The most important aspect of this software is its ability to detect when new events have occurred. Our client has, in multiple occasions, expressed the importance of accuracy. All the results returned by this software should be new events. Currently some of the results returned by the program are not exactly new events. To improve this problem our team has few areas to work on. First of all we can improve our algorithm that determines similarities of different document. In addition we can adjust our new event certainty and articles similarity threshold to avoid erroneous results. The new event certainty threshold is the cutoff point when deciding if an event is new event or not. Similarly the articles similarity threshold tells us where to draw the line when comparing two documents for similarity. Finally, improving our strength analysis algorithm will improve the overall accuracy of the software.

6.2 Table Driven Design

To improve what we have so far we plan on moving to a more table driven design. Table driven design means that any hard coded values are stored in a way they can be easily configurable. Some of the values that should be configurable are thresholds, article sources, and article categories. Table driven design is very useful for this project because there are various values that are not precisely known initially. However, our software should realize that there will be tweaking needed before finding out the best values for the variables involved. For example as we discussed above we need to change values of thresholds in order to improve the accuracy of the program. This changes can be easily made if we implement our project using table driven design. Configurable design also allows the user to get results personally relevant to them using the inputs they provide.

6.3 Weights

As we refine and improve this project one place that we need to revise how we are calculating the importance of an article. Our client has suggested few places to add weights. For example different sources should have different weight. Meaning foreign news source may have more weight local sources. Furthermore we can increase the importance of an article based on the text it contains. For example if an article contains two countries we can increase the importance of the document. This is basically the idea of ranking. Some sources are more important thus they should be given more weight. Similarly if an article contains information about important feature we define then that article should have more weight. Most of this refinement will be done in our strength analysis algorithm where we currently calculate our certainty of new event occurring and different document being similar.

6.4 Refining News Feed Sources

In order for our system to work effectively it needs to have high quality sources. One of the major refinements that needs to take place is improving the current source feeds. This can be done in several ways. One of the easiest and most obvious is to increase the total number of news feeds that our system collects. One of the potential newsfeed source that was suggested to our team was Google Alerts. Google Alerts sorts news articles by a category and provides these articles and an rss feeds which means our scraper can pick them up without an more additional work.

However, it isn’t as simple as increasing the total number of news feeds we also need to filter out if possible several of the sub rss feeds so that they no longer include social media news. Currently our system is picking up social media news as well as world events. Since, major world events happen less often than social media news the system is currently outputting more social media events then it is actual important world events. It is our hope that if we pick our rss-feeds more carefully and filter out some key words associated with social media our results will return more favorably.

Another suggestion to improve our feeds was to incorporate more foreign news feeds. The idea being that if an event is being reported internationally the system should be able to pick up on that report an event is international. If an event is international we are more sure of the event and more sure of its importance.

6.5 Additional Features

During our latest meeting with our client we demoed our project to show our progress so far. After our demo our client suggest the addition of a feature. This feature will output the most important news of the week. The purpose of this feature is to get an event with a very strong certainty that the event has occurred. If we are outputting the most important news of the week then we have a bigger data to work with. Bigger data will most likely results in a higher certainty. As we discussed earlier Accuracy is an integral part of this project. The addition of this feature will ensure that the user has the ability to get the a news with the best accuracy. Another benefit of this feature is that if we have an eventful week we may end up with numerous results. Adding a weekly news feature will further filter out some events. This will benefit users who are interested in just get the biggest news of the week. Put simply reporting the most reported event of the week will benefit two type of user; one who is concerned about accuracy and the other who is interested about the biggest news only.

6.6 Improved Classifiers

Currently our classifiers are considerably weak. All it does is look to see if the description or title contains any of the predefined words. However that is typically not good enough and doesn't always work. A good classifier is variable and adjusts with the training data given. Creating the training data is a long and tedious task that requires someone manually going through and classifying the data in our case reading a document and determining the type of event. However it seems our client has had some experience with classifiers and may have some training data to give to us so that we can accelerate the process of creating training data.

6.7 Setting upper limit for news event list

Our client would like to minimize the final list of new news event to be no longer than 20 news events. To achieve this we will generate the final list of news events but if the list is greater than 20 news events we can slightly increase the threshold used to determine if documents are similar enough to each other which will decrease the final number of news events.

7 Refinements 2

During a meeting with our client a request for displaying the final new news event list on a public website was made. To accomplish this we used several new technologies including Node.js, heroku, javascript, and D3.js. In the following sections we will talk more in depth about what does done to meet this request.

7.1 Node.js for web server

To create a simple web server we learned to use Node.js a popular javascript framework. The web server setup file have been added to the top directory as well as to a new folder "website/" viewable after logging into bitbucket website. The file "web.js" serves as the main express application that renders all of the javascript, html, and css in the file “website/new_event_list.html”. Using the popular Node.js Express framework we can serve all of the content in the “website” directory.

7.2 Heroku for deploying web server

To display the website on a public viewable URL we deployed our web app to Heroku. Heroku was chosen because it provides a free server for web app deployment. Using Heroku’s dashboard and command line tools we deployed the code in the “website” folder to https://nned.herokuapp.com/.

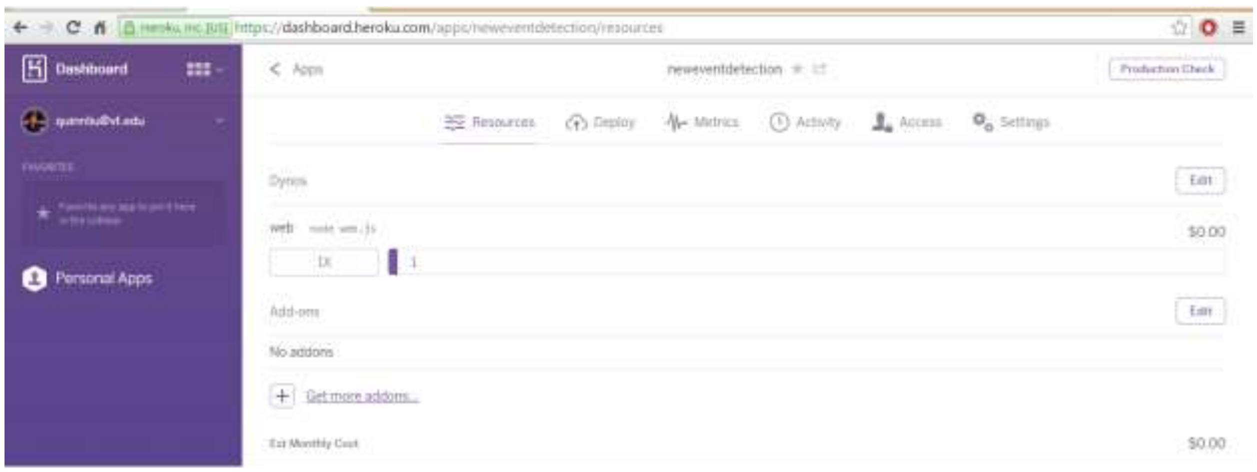


Figure 6: View of Heroku’s dashboard of the file “web.js” running inside the Heroku dyno.

7.3 D3.js and Bootstrap to display contents of file

To display the final events list there are many complications. The pipeline outlined in section 3 system architecture will output a final new news event list in a file called “final_news_events.csv” viewable here: https://github.com/quinnliu/neweventdetection/blob/master/website/final_news_events.csv.

Using regular javascript libraries we are unable to read the file on the server. Additionally, if we created a button to select a file to read the user will not be able to select the file on the ec2 instance. However, using D3.js we can read in a local .csv file and display it's contents. D3.js is a open source and popular visualization framework using Javascript to create HTML and SVG elements details here https://github.com/mbostock/d3.

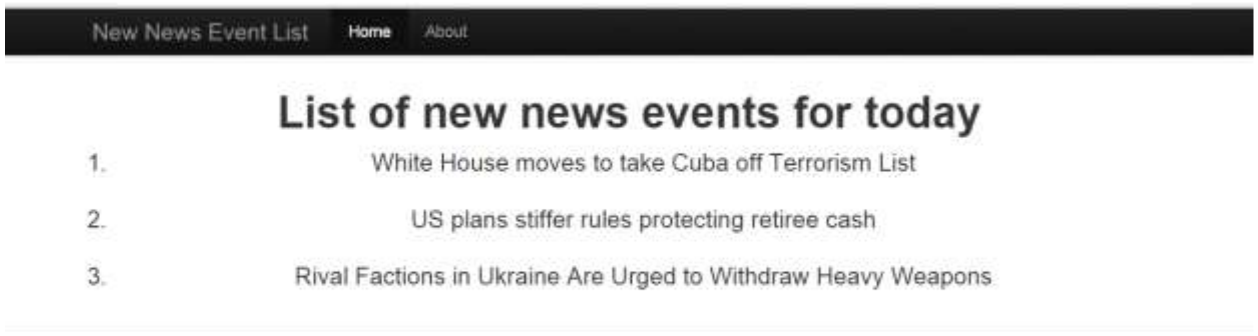
D3.js allows you to generate html elements such as the "<ol></ol>" ordered list elements and then add "<li></li>" individual list elements using it's API. Using this we can use D3.js to read the file "final_news_events.csv" and then generate the HTML code in the format shown in Figure 7.

```
<ol>
    <li>White House moves to take Cuba off Terrorism List</li>
    <li>US plans stiffer rules protecting retiree cash</li>
    <li>Rival Factions in Ukraine Are Urged to Withdraw Heavy
    Weapons</li>
</ol>
```

Figure 7: Example of HTML code that is generated by D3.js data() and enter() functions.

The power of D3.js is that no matter the number of news events in the file "final_news_events.csv" the data() and enter() methods will iterate through each line of the file and add a "<li></li>" HTML element around the news event title.

Since basic HTML and CSS is not aesthetically pleasing and our client would look at this website everyday we used Twitter Bootstrap to generate a more aesthetic pleasing website. Bootstrap is popular open source collection of HTML and CSS design templates for navigation, interface components and typography located here http://getbootstrap.com/2.3.2/. Figure 8 shows a bootstrap formated version of the final



List of new news events for today

1. White House moves to take Cuba off Terrorism List
2. US plans stiffer rules protecting retiree cash
3. Rival Factions in Ukraine Are Urged to Withdraw Heavy Weapons

Figure 8: Example of what website would look like to client

7.4 Extensions Planned

After showing the above website to the client there may be additional feature requests for links to the article and statistics about how each news article compares to others in our database. Additionally there can be buttons added to allow our client to interface with variables within our pipeline.

To automate the website to display the new news events for the day a script will need to be created to initiate our pipeline every 24 hours and then redeploy the code to Heroku to update the public website.

8 Testing

Testing is one of the most difficult portion of our project. Considering the type of system we are making the only real way to verify that we are getting the expected outcome would be to manually test it.

8.1 Testing RSS Feed Scraper

To gather the data for our system we have to collect the rss feeds at least once a day. We collect our documents every six hours. It is important that we continuously get the most recent articles so that we can detect the most recent events. That is why testing the RSS Feed Scraper is imperative. To ensure that our scraper runs every 6 hours we have set up an log that tell writes the time and date in the case that the scrapper ran into a error and the scraping was not successful. It is the job of the user of the system to check and see if there are any errors in the log.

8.2 Testing the Algorithm

It is obvious by the output that the system is working to some extent. The difficulty arises in testing the effectiveness. In order to test the effectiveness of the system we will draw upon some common techniques used in machine learning. First a training data set should be created from the articles collected in the previous stages. Then algorithm should then be trained on part of the dataset. The last step in the process is to assess the success of the trained algorithm on the remaining part of the dataset. This process can be repeated until desired results are met. We have discussed with our client and he has agreed to undertake these task as he is familiar with the techniques necessary.

8.3 Extensions Added

To make our pipeline automatic additional work was done using crontabs to automatically run parts of our pipeline. After completion of the website to display the final news events @ https://nned.herokuapp.com/ several parts of the pipeline still need to be manually run and updated to generate the final list of new news events. To automate parts of this process crontabs were used to run scripts automatically at specific times of the week. Since our code is running on an Amazon ec2 Ubuntu instance crontabs are automatically installed in all unix machines. To add a job to be run at a specific time a single line is added with specific crontab syntax followed by the list of commands to run.

```
# .----------
# | .--------
# | | .------
# | | | .----
# | | | | .--
# | | | | |
# * * * * * u
```
Figure 9: Logic of crontabs syntax

For example to run a job everyday at 6am, 12pm, and 6pm we use the crontab syntax “0 6,12,18 * * *” followed by the list of commands we want to run. The first command we run is the Rss Feed Scraper that collects all news events and stores them in our database. Then there will need to be a cronjob added to update our “Cluster/DatabaseManager.py” file lower and upper limit variables of acceptable dates of news events to consider. For example we may only want to display a final new news events between the dates 4/27/2015 and 4/29/2015. Next the main logic of our program will need to be run several minutes later giving the initial scrapper enough time to finish as it takes several minutes to run. The file “Cluster/ClusterCreationDriver.py” will then proceed to output the final list of news events. After putting this final list of news events in a .csv file format into the file “Cluster/final_news_events.csv” the final cronjob runs the git and heroku commands necessary to push the final news events to the website @ https://nned.herokuapp.com/ 3 times a day. The following are examples of cronjob commands we are using:

```
# run job everyday at 6am, 12pm, and 6pm
0 6,12,18 * * * cd /home/ubuntu/workspace/neweventdetection/RssFeedScraper; python
RssFeedScraperDriver.py

# run job everyday at 7am, 1pm, and 7pm
0 7,13,19 * * * cd /home/ubuntu/workspace/neweventdetection/;
./update_heroku_website_with_new_events_list.sh
```

8.4 Extensions Planned

To completely automate our pipeline several additional cronjobs still need to be correctly implemented. These were more difficult to implement because the files that are manually running through eclipse are difficult to run correctly as a command through the command line. While running a large .py file additional command line arguments must be passed to link to .jar files that are necessary to run the file successfully. These files are namely “Cluster/DatabaseManager.py” and “Cluster/ClusterCreationDriver.py”. To fully automate our pipeline correct cronjob will need to be written for these 2 files in the future.

10 User’s Manual

Running this system is quite simple. There are two ways to run this system. One way is to run the script manually on your machine either through an ide or command line. By running the attached scripts, you will be able to connect to the database and the results should appear in the console. The second way is to go to the website https://nned.herokuapp.com/and you can view the top five results there. Configuring the system is also quite simple. In the python file ClusterCreationDriver.py we have the values for weights, article similarity tolerance, and the number of events outputted to the console.

11 Developer's Manual

1. Install the RssFeedScraper and run it daily as a cron job. The MYSQL_CONFIG can be modified to target the desired database. The RssFeedScraper will target all urls defined in input.txt.

2. Once the database is populated run the ClusterCreationDriver. This will take a while. Upon completion a list of world events from the selected days will be output.

3. The days to select from can be changed by modifing the upper limit and lower limit defined in the DatabaseManager.

9 Bibliography

1. http://home.adelphi.edu/~siegfried/cs480/ReqsDoc.pdf
2. https://www.youtube.com/watch?v=FfLo5OHBwTo