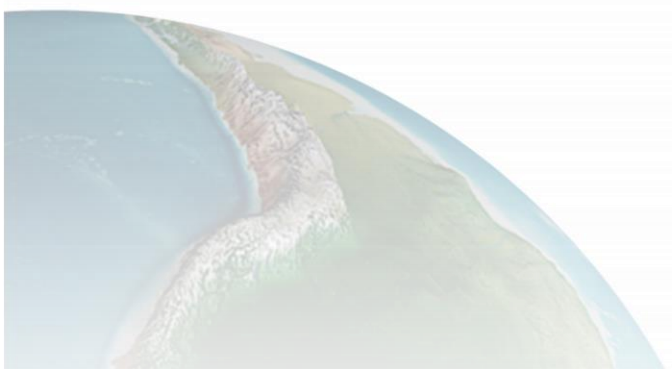
بسم الله الرحمن الرحيم

The Global Daily Journal

By

Alaa Nagaty
Reem Nasser
Ghada Ibraheem
Yara Fahmy
MennatAllah Ward

SUPERVISED BY,
PROF. DR. YOUSRY TAHA



CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF EQUATIONS

1 ABSTRACT

The aim of our Senior Project is to create and implement a news aggregator web application with more classification features of news articles and an attractive user interface. We wish to capture the attention of the people who lost interest in day-to-day World News through an interactive, explicit and appealing map interface, hence the name the Global Daily Journal.

This project involves 4 main phases. We first gathered news articles from several trusted news providing websites by parsing their corresponding RSS Feeds and storing data related to every article such as, article title, description, URL and timestamp. Each article is then tagged with a list of locations mentioned within its title and description using an open source Geo-tagging tool called CLAVIN and GeoNames' geographical database to resolve the coordinates of the locations extracted. Articles are then further classified according to defined categories; Politics, Sport, and Business. Once a geo-tag and category has been linked to each article, we displayed the number of news concerned with every location using numbered pins on Google Maps API embedded in our website. The user can click on each pin to display the list of news articles concerning that particular location and can filter the news according to category. Finally, the database is updated periodically with every news release.

2 INTRODUCTION

The World is always bustling with events. With every passing second, a fresh headline comes up in the news. Thanks to modern technology, people can be notified with the latest via our handheld gadgets.

The Global Daily Journal is a web platform that displays World News from different sources that is updated on a daily basis. Users can connect to global events happening in any place in the World from their Personal Computers in not time.

While traditional news platforms have lost audience, online news consumption has been undergoing major changes as well. Therefore, with Global Daily Journal, we wish to benefit from this by creating an online platform for World News.

The GDJ makes the news more reachable, by visualizing our World as a small village in a good user interface.

2.1 MOTIVATION

Our generation in Egypt has lost interest in World News. Owing to the turmoil in the Middle East, we have become too concerned with news in the Middle East and North Africa and have shown little if any attention towards International News. If only we could see things from a wider perspective; a viewpoint that can give us access to the stories changing our world, not only our region.

Figure 2-1 We want to see what is happening in our World, from a wider perspective

During the mid-2000s, the Egyptian nation was mesmerized with football events relating to the African Cup of Nations. We all had patriotic pride because we finally excelled in something. We showed little attention to the rest of the world. Consequently, we had little knowledge of the political, economic and environmental changes in remote countries.

Such information could help many people of different fields, ages and backgrounds. They could serve us educationally, by suggesting ideas for developments in our financial infrastructure, or political legislations or eco-friendly measures. They could also warn us from a foreseeable danger, such as a hurricane relocating from one region to another. Vast businesses can easily capture statistics relating to different countries. There are numerous benefits to information relating to Global News, but unfortunately, our community was not well-informed.

This lack of knowledge prolonged until we were introduced to Social Media such as Facebook and Twitter. Everyone became globally interconnected. Through Social Media, we gained

knowledge of the uprising in Tunis. We became politically active and started organizing protests ourselves. Our interest towards the systems of other democratic regimes in different countries, has ultimately reduced our knowledge gap. A simple counter of the trending hashtags on Twitter opened more topics for discussions among our local community. A shared article on Facebook can grab one's attention to other unknown sources of information.

However, we still do not know the top stories that reached the Global headlines today; yesterday; the past week. As much as Social Networks have kept us connected, they haven't shown us entirely the events occurring in each corner of the planet. There is no interface available that can get rid of communication barriers. That is why we wish to generate a platform that visualizes the News in every country and every day.

The Global Daily Journal must be a great way to explore the latest headlines on a daily basis. We wish for the user to take a journey across the continent, investigating the stories that change our World and our future. A faster approach to learning news off the web, and an eye-catching interface for everyone to enjoy using. We wish to create a database that covers the entire world with the different International News Agencies reporting every headline. We want our community to be connected to the world.

2.2 SURVEY

We have witnessed that during the 21st Century, not so many people check the news, whether it be on the Television, Internet or through our smartphones. According to the Pew Research Center for the People and Press, the percentage of Americans who say they regularly check the news fell 22% from 1993 to 2004 [1]. There are several reasons behind this downfall. First of all, a lot of people have been preoccupied in day-to-day matters more than they ever used to be. Due to enhanced Social Networks, we are always interconnected to the stories that are related to our circle of friends and family.

Others regard news headlines as toxic to one's health. In the past few decades, the fortunate among us have recognized the hazards of living with an overabundance of food (obesity, diabetes) and have started to change our diets. But most of us do not yet understand that news is to the mind what sugar is to the body. News is easy to digest. The media feeds us small bites of trivial matter, tidbits that don't really concern our lives and don't require thinking. That's why we experience almost no saturation. Unlike reading books and long magazine articles (which require thinking), we can swallow limitless quantities of news flashes, which are bright-coloured candies for the mind. Today, we have reached the same point in relation to information that we faced 20 years ago in regard to food. We are beginning to recognise how toxic news can be [2].

In addition to that, people prefer to check their news from online and digital sources. Online and digital news consumption, continues to increase, with many more people now getting news on cell phones, tablets or other mobile platforms. And perhaps the most dramatic change in the news environment has been the rise of social networking sites. The percentage of Americans saying they saw news or news headlines on a social networking site yesterday has doubled – from 9% to 19% – since 2010. Among adults younger than age 30, as many saw news on a social networking site the previous day (33%) as saw any television news (34%), with just 13% having read a newspaper either in print or digital form [3].

Fewer Reading, Writing on Paper

% who did this yesterday ...	2002 %	2006 %	2012 %	02-12 Change
Read a print newspaper	41	38	23	-18
Read a print magazine	23	24	17	-6
Read a book in print	34	38	30	-4
Wrote or received a personal letter	--	20	12	--

PEW RESEARCH CENTER 2012 News Consumption Survey. Q9, Q11, Q28, Q30, Q37f.

Many Read Leading Newspapers Digitally

Based on regular readers of ...	Read mostly in .... Print %	Computer/Mobile %	Other/DK %	N
New Yorker, Atlantic, Harpers	72	23	4=100	103
Economist, Bloomberg Busweek	55	37	8=100	111
Wall Street Journal	54	44	2=100	142
USA Today	48	48	4=100	127
New York Times	41	55	5=100	174

PEW RESEARCH CENTER 2012 News Consumption Survey. Q90. Based on regular readers. Figures may not add to 100% because of rounding.

Figure 2-2 New evidence indicates there is a shift in reading, from newspaper to electronics [3]

In spite of an expanding variety of ways to get news, a sizable minority of young people continues to go 'newsless' on a typical day. Fully 29% of those younger than 25 say they got no news yesterday either from digital news platforms, including cell phones and social networks, or traditional news platforms. That is little changed from 33% in 2010.

Older Americans are less likely to go newsless: 19% of those between 25 and 39, and smaller percentages of older age groups, say they got no news yesterday. These figures have changed little over the years.

Young people also consistently spend less time with the news than do older Americans, which is in part attributable to the relatively large share that gets no news on a typical day. In the current survey, those younger than 30 spent an average of 45 minutes getting news yesterday. Older age groups spent an hour or more with news, on average, with those 65 and older spending an average of 83 minutes with the news yesterday. Age differences in time spent with the news have changed little since the 1990s. [4]

Young People Continue to Spend Less Time with the News

	Average total minutes yesterday									
	1994	1996	1998	2000	2002	2004	2006	2008	2010	2012
Total	74	66	65	59	59	72	69	66	70	67
18-29	56	44	48	42	38	45	49	46	45	45
30-39	69	60	53	50	57	70	65	63	68	62
40-49	75	65	65	58	56	73	64	67	74	71
50-64	83	79	69	64	71	82	76	74	81	76
65+	90	88	96	80	81	88	79	84	83	83

PEW RESEARCH CENTER 2012 News Consumption Survey. All averages are estimated based on total time spent watching TV news, reading a print version of the newspaper, listening to news on the radio and getting news online, including online/digital versions of newspapers. Online news added in 2004. In 2004 and earlier, all newspaper reading is assumed to be in print.

Figure 2-3 Young people spend less time compared to older age groups [4]

Aside from the means of news delivery and the age group interested in the latest news, several foreign news stories attracted high levels of public interests. The dramatic changes unfolding in several Middle Eastern countries drew considerable public attention. In February 2011, nearly four-in-ten Americans (39%) followed anti-government protests in Egypt and the resignation of Hosni Mubarak very closely. And even before the United States and its allies

launched airstrikes in support of anti-government rebels in Libya in late March, 38% tracked news about violence in Libya very closely [5]. However, once a piece of news update takes a monotonic turn, interests begin to decline.

For Public, No Breakthrough Foreign Stories So Far in 2012

Top foreign stories…	Very closely	Less closely	DK
2012	%	%	%
Cruise ship accident off coast of Italy(*Jan*)	30	70	1=100
U.S. soldier in Afghanistan accused of killings (*Mar*)	28	71	1=100
Iran-Israel tensions (*Feb*)	26	75	*=100
Undercover agent in Yemen foils plane plot (*May*)	24	75	1=100
N. Korea's failed rocket launch	22	77	1=100
U.S. rescue of aid workers in Somalia (*Jan*)	21	78	1=100
European economic problems (current)	*17*	*83*	*1=100*
Political violence in Syria (current)	*12*	*87*	*1=100*
2011			
Japan earthquake and nuclear disaster (*March*)	55	44	*=100
Osama bin laden killed (*May*)	50	49	1=100
U.S. troops killed in copter crash in Afghanistan (*Aug*)	39	60	1=100
Anti-government protests in Egypt (*Feb*)	39	60	1=100
Violence in Libya (*Feb*)	38	61	*=100
Libya air strikes (*Apr*)	37	62	1=100
U.S. troop withdrawal from Iraq (*Dec*)	34	64	2=100

Figure 2-4 Statistics on World News interests on particular headlines [5]

All in all, we have concluded from the statistics mentioned in this survey that most people prefer receiving the latest news updates from Social Media, but social media does not fully inform its users with every aspect of foreign news. Those who consistently follow the news through the newspaper/website and/or Television are well informed and fit in an older age group compared to those who receive the updates via Social Media. Simultaneously, foreign news has sparked interests of many people and may have also added more to their knowledge of remote communities.

2.3 RELATED WORK

Listed below are a number of commercially distributed projects that share similar aspects to the website we have in mind. They stand as a source of inspiration and give us ideas to define our scope.

2.3.1 BBC Most Popular Now

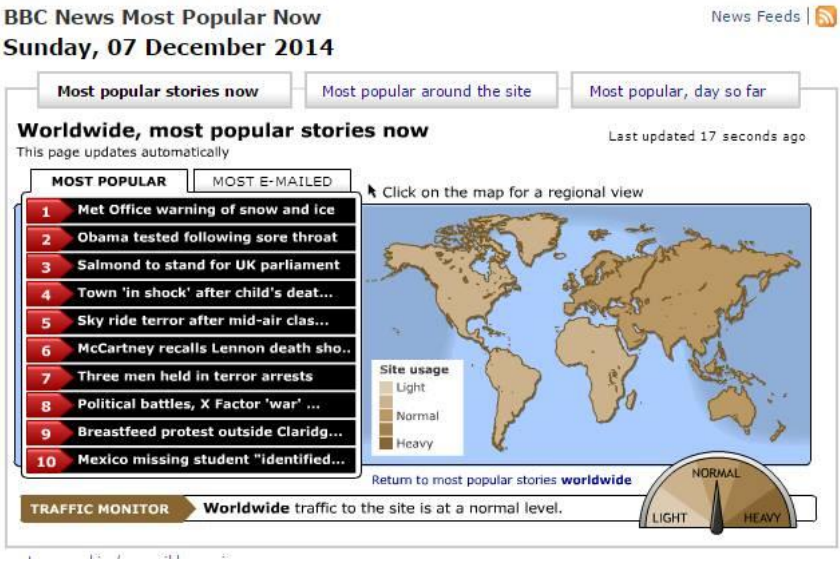


Figure 2-5 BBC Most Popular Stories Now [6]

- This application was developed to grab user's attention to the website and give the website more interactivity.
- This system is designed to monitor user's interests of website articles by feeding the BBC Live Stats system with World News.
- It creates a real time system that is updated automatically, with every passing minute.
- The application divides the world into regions and finds out the popular stories read by BBC News website users across those regions.
- They promise protection of user's privacy, so they didn't store any personally identifiable information, for that they use geographical IP lookups. Geographical region data is based on IP address using a 3rd party ge-olocation service. IP addresses are immediately converted to the matching country and the IP data is deleted. And data is only kept in the system for a maximum of 24 hours.
- The system is visualized as a world map, the map can be filtered by continent to show the top 10 'most popular' and 'most emailed' BBC News articles.
- There's also a speedometer indicating how traffic to the site is compared to 'normal' levels, it compared with the recent traffic average for that time of day calculated from the BBC News Live Stats system that measures the live news consumption and indicates whether it is a quieter or busier day than average [6].

2.3.2 Rich Site Summary (RSS)

Rich Site Summary is a technology which allows you to easily stay informed by retrieving the latest content from the sites you are interested in. You save time by not needing to visit each site individually. You ensure your privacy, by not needing to join each site's email newsletter. It allows you to see when websites have added new content. You can get the latest headlines and video in one place, as soon as it's published.

2.3.3 **Twitter**

- Geo-tagging is the process of adding geographical information to various media in the form of metadata. The data usually consists of coordinates, latitude and longitude, but may even include bearing, altitude, distance and place names. It can be used to find location-specific websites, news and other information. It is based on positions and coordinates and is often directly taken from a global positioning system (GPS) [7].
- Twitter Geo-tagging is simply attaching your exact location to an individual tweet.
- Given a point, the API returns the boundary information: neighbourhood, city, etc. Twitter recently integrated the service with its front-end to let user's geo-tag an area as opposed to a specific point [8].
- There are numerous websites and application that search specific cities to find local tweets. These tweets are often displayed as content on their websites. A tweet that is geo-tagged to that location will appear in that search. In this way, tweets can be broadcasted to a small region [9].

2.3.4 **Google News**

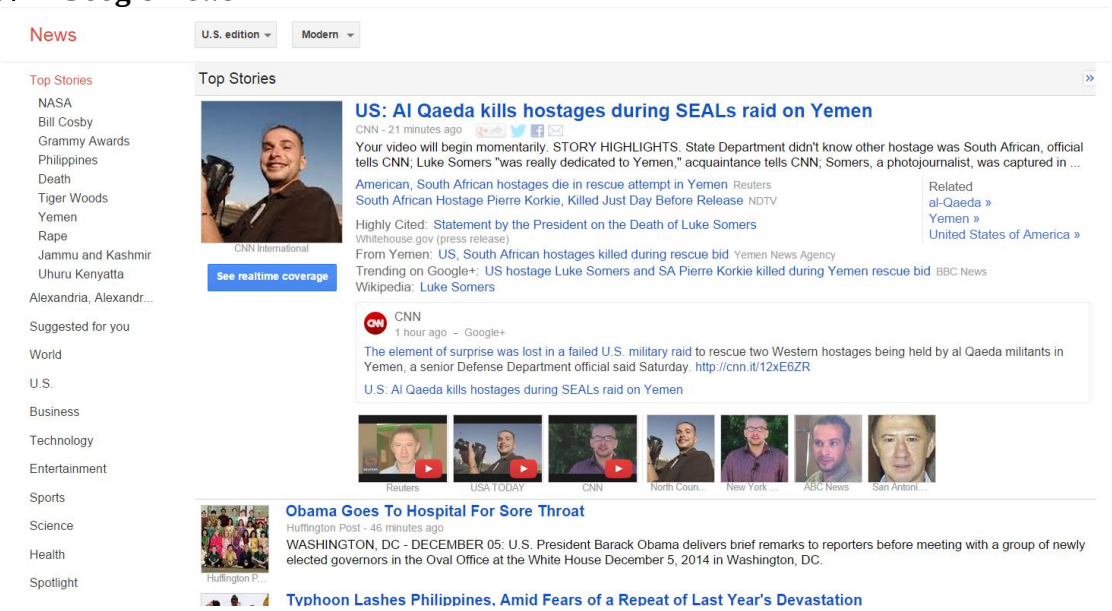


Figure 2-6 Google News Homepage [10]

- It is a computer-generated news site that aggregates headlines from news sources worldwide, group's similar stories together and displays them according to each reader's personalized interests [10].
- It generally recommends content to readers via an intelligent algorithm primary based on how long users spend reading articles.
- Articles are ranked based on originality, freshness, quality, expertise of source and whether a lot of other sources around the Web are pointing to a particular article.
- The algorithm Google News uses, favours plain and simple articles. The longer the story, the greater expense to cut, paste, compile and mangle abstracts of the story produced by media outlets.
- The algorithm also records the usage patterns. Links going from the news search engine's web page to individual articles may be monitored for usage (e.g., clicks). News sources that are selected often are detected and a value proportional to observed usage is assigned. Well known sites, such as CNN, tend to be preferred to less popular sites [11].

2.3.5 The Global Database of Events

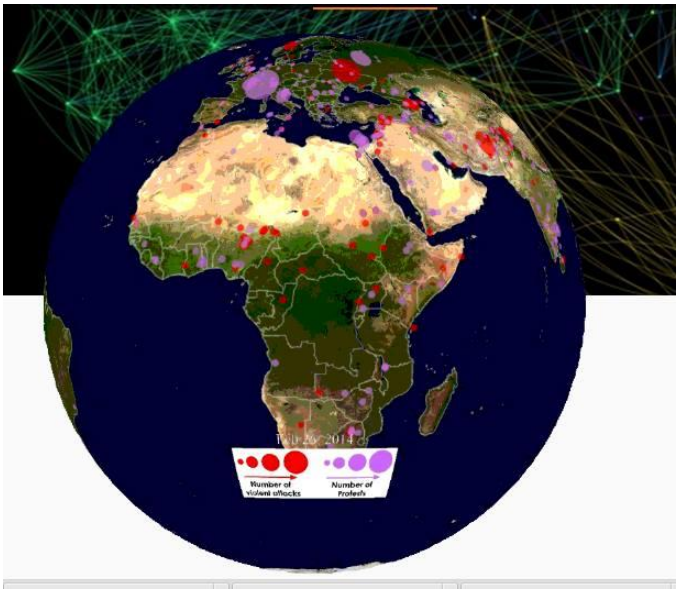


Figure 2-7 GDELT's visualisation of Global Activity [12]

- Its goal is to construct a catalogue of human societal-scale behaviour and beliefs across the world, connecting everyone, organization, location, news source, and event across the planet into a single massive network that captures what's happening around the world, what its context is and who's involved, and how the world is feeling about it, every single day.
- They have been collecting and documenting the news about events related to conflict and political protest dating from 1979 until 2013. GDELT gathering new data through the various global news services, and automatically updating every day with a massive network diagram connecting every person, organization, location and theme to this event database.
- The interface of the database is a map of the globe showing pulsing spots at the regions of the news with two different colours according to the nature of the news, a red coloured spot indicates news about "*violence against civilians*" while a purple coloured spot indicates news about "*Protests*". The spots vary in size according to how important the news are and how many news corporations have covered it.
- In a single map you are seeing an overview of major global activity each day, as captured by the world's news media and managed by the database of events. All protest and conflict events are grouped together by city/location. For the animated map layer, if a location has both protest and conflict events, it is coloured by whichever there are more of. For the daily map layers, dots are displayed separately at each location for protests and conflict and are sized based on the volume of coverage given to that type of event at that location. Thus, locations with more "important" events will be displayed using a larger dot to indicate major evolving situations [12].

2.4 EXCLUSIVITY

The listed projects have many common features. We wish to combine the features that are uncommon in each project into one web application. Most news websites specify regions according to continents. On the other hand, we would like to give attention to smaller areas on the globe, like cities and towns. To our knowledge, there is no news aggregator that lists all the news on an interactive map. This could revolutionize the concept of news aggregation.

2.5 SCOPE

In Summary:

- The interface of the project will be a world map of the globe to make it faster for the user to surf through different regions.

- The map will include pins to mark the presence of news anywhere in the world. The number of news concerning a location will be indicated on the pin. The User must click on the pin to see all the stories in that region.

- We will be using reliable News Sources, such as BBC NEWs, CNN, AlJazeera International and etc.

- The news will be updated on a daily basis, so that the user is updated with the latest news each day.

- The User can also zoom in a Continent or Country to narrow the pins displayed all the way to Cities.

- Also categorize the type of news according to specific categories. The category is displayed on a side panel. Each category listed, shows a different set of Pulses.

3 ANALYSIS

Having set our objectives and scope of work we intend to achieve, we now proceed on analysing all the possible tools that could help us collect important news stories from a variety of sources, classify them and introduce an interface that will be attractive to the users. Thereby, we divide our project into 5 main tasks:

- How to extract the news from various sources
- How to tag the news with their corresponding locations
- How to classify them according to category
- How to refresh the database

3.1 NEWS EXTRACTION

Web news article contents extraction is vital to provide news indexing in our project. Most of the traditional methods need to analyse the layout of news website pages to generate the wrappers manually or automatically. It is a costly work and needs much maintenance during the extraction over a long period of time, and consistently changing website layouts. Consequently, we decided it is better to construct an automatic Web news article contents extraction system based on RSS feeds. It is far more effective and efficient to extract the news articles without the analysis of every news site before extraction. RSS Feeds are applicable to general types of content organization and independent of news page layout.

Generally, other content extraction methods include two steps. Initially, the news websites are crawled to collect the news pages. Secondly, the news article contents are extracted from news pages. However, the news sites compromise of Web pages with different content. Aside from the pages that include news articles, there are many non-news pages, such as blog, shopping, weather, forum, and yellow pages. Furthermore, these news pages are scattered in different sections of news sites. The news sites are crawled to find as many news pages as possible, but in reality, it is difficult to recognize and acquire all the news pages quickly from a large number of Web pages.

News have other foreign elements such as advertisements, related stories and comments. In order to recognize and extract the news article segments from the rest of the page, wrappers are generated based on the analysis of the page's web layout. The web page layout is the style and design in which text and/or pictures are displayed and distributed on a website. Not only do various news pages have different web layouts, but some news websites consist of more than one layout. As a result, traditional extraction methods are forced to extensively analyse the news page layout prior to content extraction. In addition, websites update their layouts on an irregular basis. Any layout update means re-analysing the new layout. All these factors has an adverse effect on time and performance of this phase.

Under these circumstances, we decided to parse news-related data through RSS feeds. As mentioned earlier in the Introduction (pg. 11), RSS is a family of Web feed formats used to publish frequently updated content such as news headlines. We can easily collect the latest news from news RSS feeds as soon as they are published. There will be no need to use machine learning methods to extract content since RSS feeds are independent of web layouts and do not require pre-analysis of the layout or maintenance. The method is proven to be applicable and effective on the long run. This gives us the opportunity to increase the number of news sources we can extract information from.

An increasing number of news websites are now embedding RSS feeds to allow easy access of news through subscriptions. RSS is an XML-based format for sharing and distributing frequently updated Web content such as news and blogs. A news RSS document, generally

called a *news RSS feed*, includes headlines, summaries and links to the news article page. RSS makes it possible for people to keep up to date with their favourite websites in an automated manner. In conclusion:

- More and more news sites embed RSS feeds in their websites. As a matter of fact, more than 97% of America's top newspaper websites provide RSS feeds [13]. This eases the process of news extraction and widens our list news sources.
- Extracting content from RSS feeds is far better than extracting them through machine learning methods.
- An RSS feed news extractor can be developed easily and there is no need to maintain the code with every web layout change [14].

3.2 LOCATION EXTRACTION AND GEO-TAGGING

Location extraction is a method of *Entity Extraction*; a semantic technology that tries to find a meaning in unstructured text. An *Entity Extractor* promotes words in text to concepts; this is typically realized in the form of *entity tagging*, where an ontology is associated with a word or phrase, such as, PERSON, PLACE, and ORGANIZATION etc. In our project, we will associate every object with PLACE.

Once entities have been 'tagged', the next step is to 'resolve' them to a global concept or entity. This step is known as *Entity Resolution*. Geo-tagging is considered a more sophisticated form of entity resolution with techniques that include associating geographic coordinates. For instance, we not only want to know that New York City is a LOCATION, but also, that it's center latitude and longitude is 40.7142° N, 74.0064° W, the location is in the "New York State" administrative district, and in the country of the United States of America.

Unlike Twitter, news RSS feeds do not have automatically set tags that define the location of the article. Therefore, we need to develop a tool that sets geographical tags to articles as well as linking it to the locations' corresponding longitude and latitude to map it on our website. For years the Geo-tagging market has been dominated by a very small number of commercial products; many entity extractors can identify locations, but few actually resolve that location to a fixed point in space. For that, they need a geospatial dictionary to be incorporated in their product.

Fortunately, Beirco Technologies have provided an open source software application for document geo-tagging and geo-parsing which is distributed under the name *CLAVIN* [15]. CLAVIN stands for Cartographic Location And Vicinity INdexer; It automatically extracts location names from unstructured text and resolves them against a *gazetteer* to produce data-rich geographic entities. It's fast, accurate, and scales to accommodate big data in the cloud.

The gazetteer is brought from GeoNames gazetteer data. It is both the world's largest and most used gazetteer. The data is based on freely available national gazetteers and datasets, as well as volunteered geographic information (VGI).

CLAVIN combines various open source tools with natural language processing techniques to extract and resolve geospatial entities from text, intelligently, accurately, and automatically. Its *named entity recognition* handles and distinguishes alternate names, while *fuzzy matching* is used to capture misspelled location names, including phonetic spelling and typographical errors. CLAVIN also recognizes alternate names for the same entity (e.g., "Ivory Coast" and "Côte d'Ivoire"), and intelligently disambiguates between ambiguous location names. Take, for example, the following block of text:

"I visited the Sears Tower in Chicago only to find out there were exciting attractions in Springfield. After Springfield,

Chuck and I drove east through Indiana to West Virginia, stopping in Harper's Ferry. We finally made it to our destination in Washington, DC on Tuesday."

A reader or a simple extractor can easily list the locations mentioned (Chicago, Springfield, Indiana, West Virginia, Washington and DC), but would not be able to locate all of them on a map. According to Wikipedia, there are almost 70+ regions with the name Springfield across the globe, most of them located in the USA. With CLAVIN, however, it will accurately identify it as Springfield the city in the state of Illinois with the coordinates (39°48'06.2"N 89°38'37.4"W) since the writer explicitly mentioned their journey around Chicago, the capital of Illinois.

By enriching documents with structured geo-data, CLAVIN enables advanced geospatial analytics on massive volumes of unstructured text without massive cost implications. Architecturally, CLAVIN is extremely simple. CLAVIN was written in Java, and can be bundled in a Java Web Application as a web service allowing our project to access it [16].

The accuracy of its output is proven to be around 75%. Despite its intelligent algorithm, GeoNames Gazetteer may have missing and incorrect data. The figure below illustrates a visualization of the density of place names listed in the GeoNames Gazetteer [17].

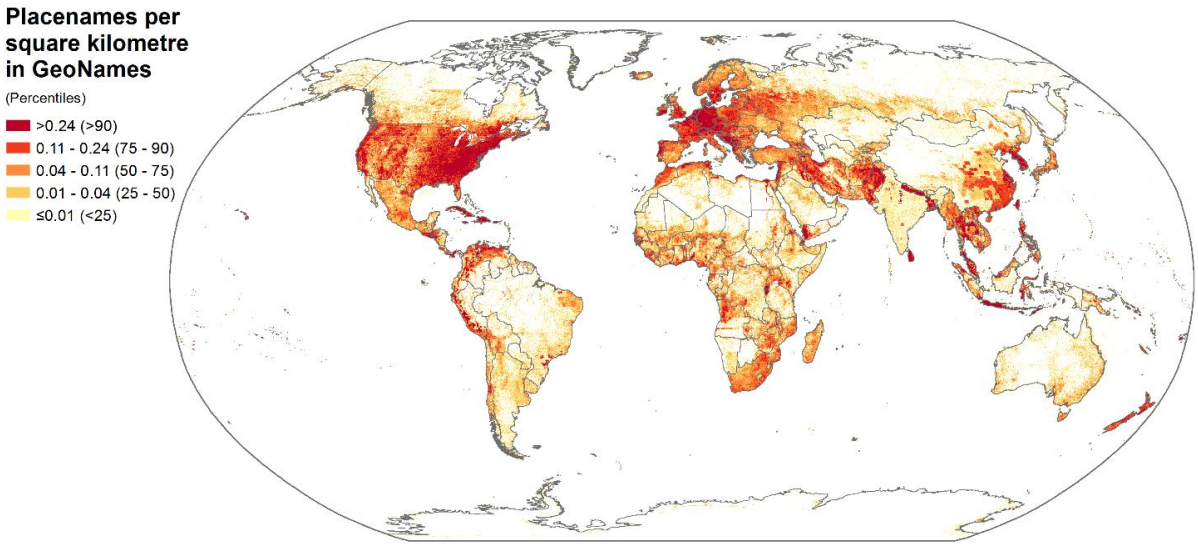


Figure 3-1 Placenames per square kilometre in GeoNames Error! Bookmark not defined.

Surprisingly, the clusters that we see are characterized by unusual patterns. Not only do we see the concentrated data in Central Europe and the United States with large amounts of geographic information, but we also see significant densities in places like Sri Lanka, Iran, and Nepal.

It is clear how national and international policies play a large role in the construction of this gazetteer. The United States is by far the most representative country in the dataset, accounting for a quarter of the total number of placenames. Nepal comes in 11th, apparently thanks to a project funded by the European Union in 2001, and counts more place names than India and the UK put together.

On the other hand, Iran counts almost the same number of placenames as Germany. North Korea is described in almost as much detail as Austria, and Sri Lanka counts even more placenames. It is likely that some of these patterns exist due to the fact that the National Geospatial-Intelligence Agency's (NGA) and the United States Board on Geographic Names are the sources of many place names outside the United States and Canada. Created in 2003 as

part of the U.S. intelligence, the strategy of the NGA is to provide *"support to military and intelligence operations, intelligence analysis, homeland defense, and humanitarian and disaster relief".* These objectives might explain why we see a particular focus on places like Iran, North Korea, and Sri Lanka [18].

Not only does CLAVIN have a really high accuracy in comparison to other extraction methods, but it is also astoundingly fast. News get updated on an hourly basis, so we will be dealing with large amounts of data. CLAVIN can resolve 100 locations in a second per CPU. It can process 1 million documents containing 5.7 million locations in under one hour on a 9-node Hadoop cluster. At this rate, we can tag a single article in less than a second.

3.3 CATEGORIZATION

Text mining and classification remains an ongoing research in the fields of machine learning. Its complexity underlies in the discovery of new, previously unknown information, by automatically extracting information from different written resources. It lies at the intersection of IT, mathematics, and natural language. It is primarily concerned with three topics:

- **Classification.** Assigning items to arbitrary predefined categories based on a set of training data of similar items
- **Recommendation.** Recommending items based on observations of similar items
- **Clustering.** Identifying subgroups within a population of data

A key element is the linking together of the extracted information together to form new facts or new hypotheses to be explored further by more conventional means of experimentation.

Text mining offers a solution to this problem by replacing or supplementing the human reader with automatic systems Undeterred by the text explosion. It involves analysing a large Collection of documents to discover previously unknown Information.

There have been variety of supervised learning techniques that have demonstrated reasonable performance for document classification. Some of these techniques includes k-nearest neighbour, support vector machines, boosting and rule learning algorithms.

For this project, we focus on Naive Bayes classification, Maximum Entropy classification and probabilistic grammar classification.

3.3.1 Naïve Bayes Classification

Bayesian classifiers are probabilistic approaches that make strong assumptions about how the data is generated, and posit a probabilistic model that embodies these assumptions. Bayesian classifiers usually use1supervised learning on training examples to estimate the parameters of the generative model. Classification on new examples is performed with Bayes' rule by selecting the category that is most likely to have generated the example.

The Naïve Bayes classifier is the simplest of these classifiers, in that it assumes that all features of the examples are independent of each other given the context of the category. This is the so-called "Naïve Bayes Assumption". While this assumption is clearly false in most real-world tasks, Naïve Bayes often performs classification very well. Despite this, practical applications of Naïve Bayes classifier have had high degrees of accuracy in many cases.

In the case of document classification, number of features is document classification is usually proportional to the vocabulary size of the training document set. This number can be quite large in many cases so the Naïve Bayes classifier is a major advantage over other classification techniques.

3.3.2 Maximum Entropy Classification

Maximum entropy has been widely used for a variety of natural language tasks, including language modelling, text segmentation, part-of-speech tagging, and prepositional phrase attachment. Maximum entropy has been shown to be a viable and competitive algorithm in these domains.

Maximum entropy is a general technique for estimating probability distributions from data. The core principle in maximum entropy is that nothing should be assumed about the probability distribution other than observations during supervised learning i.e. the distribution should be as uniform as possible. A uniform distribution has the property of maximal entropy.

Labelled training data is used to derive a set of constraints for the model that characterize the category specific expectations for the distribution. Constraints are represented as expected values of "features", any real-valued function of an example. A variety of hill-climbing algorithms are available to find the maximum entropy distribution that is consistent with the given constraints.

For document classification, maximum entropy estimates the conditional distribution of the category label given a document. Features are defined for each document. The labelled training data is used to estimate the expected value of these features on a category-by-category basis. The hill-climbing algorithm finds a text classifier of an exponential form that is consistent with the constraints from the labelled data.

3.3.3 Probabilistic Grammar Classification

Probabilistic statistical parsers use information from a tagged training set to create a model for the structure of the grammar in a language. They look at empirical data of what part of speech follows what part of speech in the training data, and can thus make predictions on what it expects to see as it parses sentences whose words are already tagged with their part of speech.

One important observation to make about this method compared to the previous ones is that it depends more on the structure of the language, and not on the language itself. For example, the actual content of the words in the sentence matter less than their part of speech in some areas, so this method is looking for ways that the grammar of different sections are structured differently.

For example, consider a news article compared to a sports article. Both articles may be written with the same style (sentence structure) because they are both reporting on the facts of an event or people. If we look at the content (as we do in the classifiers above), we may be able to distinguish them. Sports articles have words like baseball, team, game, whereas news articles have terms like class, explosion, and department.

However, if we compare news and opinion articles, their content may be the same, but the sentence structure may be different. Opinion columns try to be persuasive, so their language is more drawn out and flowery. This is information that can be captured by theses probabilistic statistical parsers that the above classifiers may miss.

3.4 DATABASE UPDATE

To figure out a fixed interval for the database to update and insert new data periodically, we first need to build up data and note the insertion time from the news extractor and geo-tagger. Based on the analysis made by CLAVIN, we guarantee a swift and fast execution and insertion time. What may worry us is the time taken to parse news RSS feeds. It is still too early to evaluate on how long it will take, but there is one thing we know for sure, that parsing will occur sequentially. It will go through each website, one-by-one. Performance-wise, this is not

very effective. Therefore, parsing the news should be a multi-threaded process. We will have to consider this when implementing this phase.

The name of the project indicates that all our data must be renewed daily similar to a newspaper distributed every morning. But news updates rush in every hour of the day, to force the user to wait for the next day and check for new information that might possibly be outdated is not a good idea. As a result, we decided that it would be best to point out a two facts:

- The entire news database will be reset every 24 hours.
- New newsflash will be inserted every fixed interval. This interval will be decided upon recording the maximum execution time of a news parser.

4 SYSTEM DESIGN

We managed to build a design of the development cycle from our analysis. Now that we have our boundaries defined and tools listed, we can draw a picture of what are the inputs and outputs of each phase and in what sequence of actions should we take.

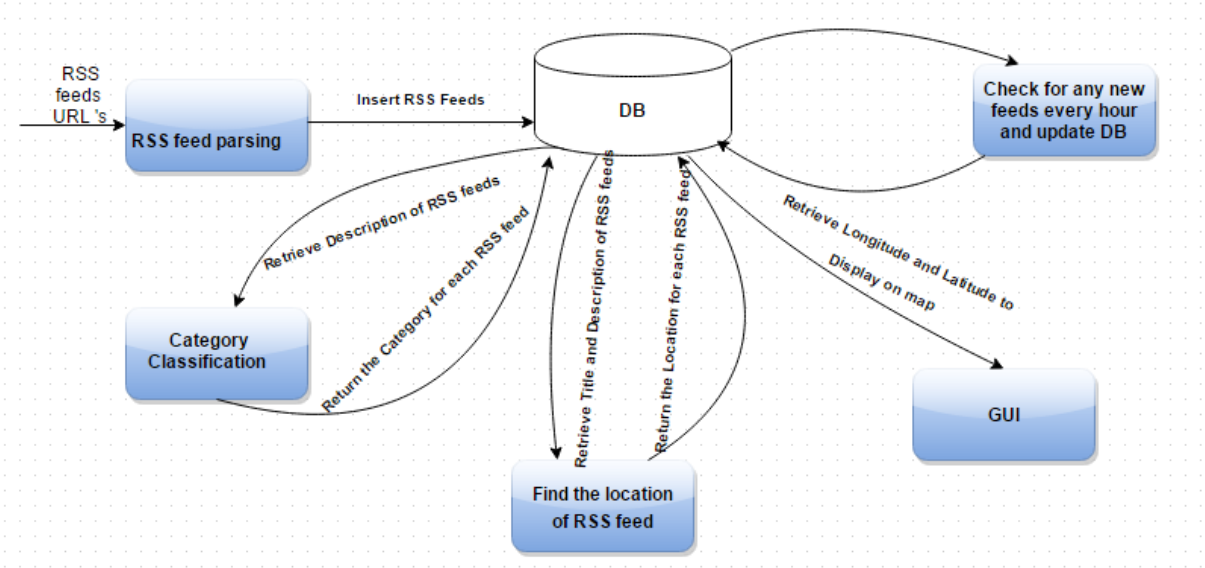


Figure 4-1 System overview

We believe that it is still too early to go through intrinsic details of the design for each phase. The design for each phase may be prone to changes resulting from the output of its preceding phase. Consequently, we decided to work on our design and implementation in an agile development plan. Each phase, will have its own design and implementation.

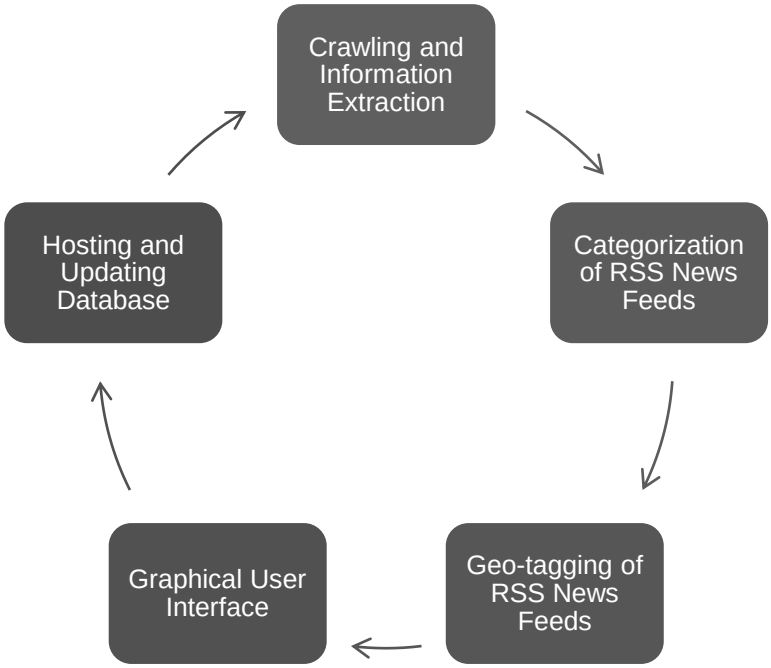


Figure 4-2 Simplified block diagram of the 5 main phases

5 COMPONENT DESIGN

5.1 PHASE1: CRAWLING AND INFORMATION EXTRACTION

5.1.1 Introduction

As previously stated, the HTML format and design in each News Site differs and constantly changes with each redevelopment and renewal, we decided to crawl and extract information from RSS Feeds. It saves us time by not taking into consideration the different HTML layouts for each site. Also the size of RSS content is small which will make it faster and easier to parse few data especially in the next two phases which depend on text mining [19].

RSS is an XML-based format that consists of designated elements that are consistent for all RSS feeds and conform to the XML 1.0 specification. These elements need to stay consistent to allow for a standardized data format that RSS aggregators can then consume.

The file holds one channel at least, this is the website that provides the information. The channel provides some articles or data. These are web pages from the same site, or from other sites [20].

Main RSS tags:

The following are the main RSS tags used as mention in [21]

- **<rss>**: The global container.
- **<channel>**: A distributing channel. It has several descriptive tags and holds one or several items.

```
<rss version="2.0">
<channel>
...
</channel>
</rss>
```

Required tags for the channel

- **<title>** The title of the channel. Should contains the name.
- **<link>** URL of the website that provides this channel.
- **<description>** Summary of what the provider is.
- one <**item>** tag at least, for the content.

```
<rss version="2.0">
<channel>
    <title>XUL</title>
    <link>http://www.xul.fr</link>
    <description></description>
    <item>
    ...
    </item>
</channel>
</rss>
```

Optional tags for the channel

- **<language>** The human language used for the text.
- **<docs>** Where to find the doc for the format of the file, may be Harvard.
- **<webmaster>** E-mail.
- **<pubDate>** Publishing date.

Items of the channel

- Each <**item>** tag must hold these tags:
- **<title>** Title of the article.
- **<link>** The URL of the page.
- **<description>** Summary of the article.

```
<item>
<title>XUL news</title>
<link>http://www.xul.fr/index.php</link>
<description>... some text...</description>
</item>
```

Some Additional Info for this Article

- **pubDate**. Publishing date.
- **guid**. A string of character that is unique to designate this item.
- **category**. The category of the article.
- etc.

5.1.2 **Design**

In this step, we will design in details the producer and algorithm that we have used in order to extract information we need from RSS Feeds and store them for further processing.

Algorithm

1. RSS feeds are parsed in order to retrieve the contents of the items tags, we only emphasize on title, description and link tags. We have choose six URLs to get the corresponding RSS feeds. In this project, the number and the scope of the URLs would be predefined as follows:
   - BBC World News: http://feeds.bbci.co.uk/news/rss.xml?edition
   - AlJazeera International: http://america.aljazeera.com/content/ajam/articles.rss
   - ABC News: http://feeds.abcnews.com/abcnews/topstories
   - The New York Times: http://www.nytimes.com/services/xml/rss/nyt/HomePage.xml
   - CNN: http://rss.cnn.com/rss/cnn_topstories.rss
   - NBC News: http://feeds.nbcnews.com/feeds/topstories

However, the system can function for any RSS feeds and is not restricted to the predefined URLs.

2. After parsing the data we need to store them in order to use them in next phases, so we have created database we be described briefly in implementation step.

5.1.3 Implementation

Below is a state diagram of the steps that will guide us to the intended outcome. The tools we will be using are explained within each procedure.
The project is implemented using Python2.7 using Canopy as the Python Editor. It uses. All tools are Python modules. Throughout this step, we will be explaining the basic of functions of each procedure in the diagram.

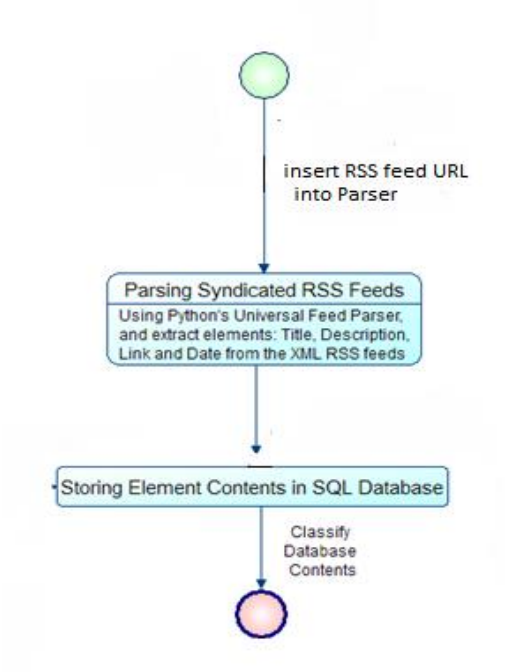


Figure 5-1 State Diagram of the first part of implementation

Getting the RSS Feeds

Our main aim in this step is to get the RSS Feeds of different news sources after we've inserted their corresponding URL links then parsing the XML files. To parse the extracted RSS Feeds, we use Python's Universal Feed Parser. It can handle RSS 0.90, Netscape RSS 0.91, Userland RSS 0.91, RSS 0.92, RSS 0.93, RSS 0.94, RSS 1.0, RSS 2.0, Atom 0.3, Atom 1.0, and CDF feeds. It also parses several popular extension modules, including Dublin Core and Apple's iTunes extensions. **Universal Feed Parser** is easy to use; the module is self-contained in a single file, `feedparser.py`, and it has one primary public function, `parse`. `Parse` takes a number of arguments, but only one is required, and it can be a URL, a local filename, or a raw string containing feed data in any format. The returned are the RSS feeds of the news source [22].

The most commonly used elements are title, link, description, publication date, and entry for our project, we only focused on title, link, and description. These tags are encompassed into a superior tag item, which must always hold these tags. In News Feeds, the each item tag specifies a headline.

```
Input = List_OF_RSS feeds_URL's
for each rss_url in Input_list
     feedparser.parse(rss_url)
     return->feeds
for feed in feeds:
  return ( feed[ items ] )
//then we can access feed[ items ].title, feed[ items
].description, feed[ items ].link
```

In the example above, we passed a list of RSS News Feeds, and parsed each feed. Parsing each feed individually takes significant time. So we improved its performance by implementing threads into our code. We used a Future class that provides a legible and intuitive way to achieve parallel parsing at once. Although Python's thread syntax is acceptable, it can still be a pain if all one wants to do is run a time-consuming function in a separate thread, while allowing the main thread to continue uninterrupted. If the Future has completed executing the parse function, the call returns immediately the element contents in an 'entries' list. The future only runs a function once, no matter how many times you read it. Thus, saving a lot of time [23].

Sorting in the Database

We have used MySQL to build our database and hosting it on our local machines using WampServer and xampp that provide a Windows web development environment. It allows to create web applications with Apache2, PHP and a MySQL database. Alongside, phpMyAdmin allows you to easily manage the database.

The system requirements are as shown:

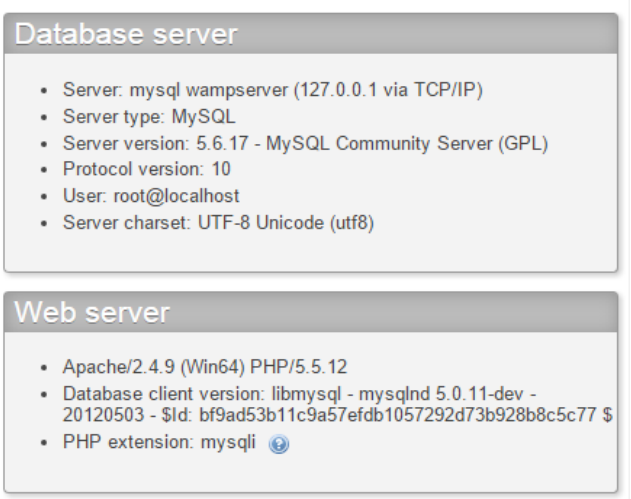


Figure 5-2 Database System Requirements

The Database is called ‘GDJ’ which represents the initials of our project name, and it’s made up of 4 entities:

- **News** (ID, Title, Link, description, time stamp, category_ID)
- **Geonames** (geoname_id, longitude, latitude, asciiname)
- **news_geonames** (News_ID, geonames_ID)
- **Category** (ID, name)

The ER-Diagram of our database design as shown below:

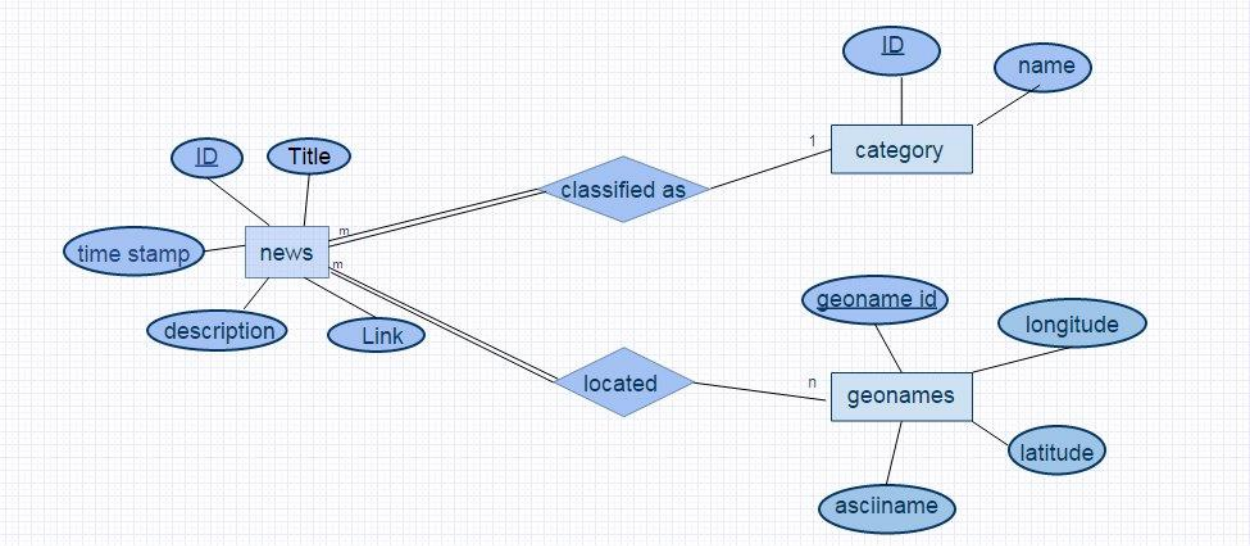


Figure 5-3 ER-diagram of Database

The Schema Diagram is as shown below:

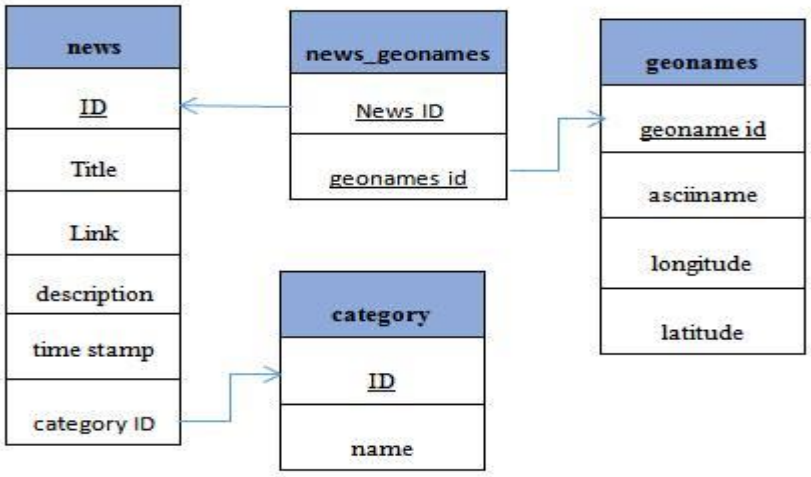


Figure 5-4 Schema Diagram

Table 5-1 Dictionary of table attributes

News Table	
ID	Unique Primary Key, Auto-increments with every new entry
Title	Article Headline
Link	URL link address of the article
Description (`description`)	Article short description of news article content
GeoNames Table	
GeoNameID (`geonames_id`)	Unique Primary for every location
Ascii Name (`asciiname`)	Name of the location
Longitude (`longitude`)	Longitude coordinates of the location
Latitude (`latitude`)	Latitude coordinates of the location
Category Table	
ID	Unique Primary Key for each category
Name (`name`)	Corresponding name of the category

After creation of database, it will be used to insert parsed tags of RSS feeds into news table, we have import Genomes database into geonames table of our database in order to be used in third phase (geotagging).the attributes that are displayed in the schema are the most important attributes that we will use.

5.2 PHASE2: CATEGORIZATION OF RSS NEWS FEEDS

5.2.1 Introduction

In order to remain updated of the latest news articles many users subscribe to various RSS feeds. However, many a times this information is scattered across various news sources and spans more than one domain. Our system provides a single RSS feed that presents all the news items from various different news sources and groups them into categories. This would save a lot of user's time which he would otherwise spend in visiting various news sites and finding top news of his category of interest.

Our project aims to process RSS feeds and obtaining a single, well-categorized output feed. The system accepts the description tag of the RSS documents have been parsed from previous phase of the different news sources acting as an input to the system. Small diagram for the general steps of this phase is shown in the figure below [24].

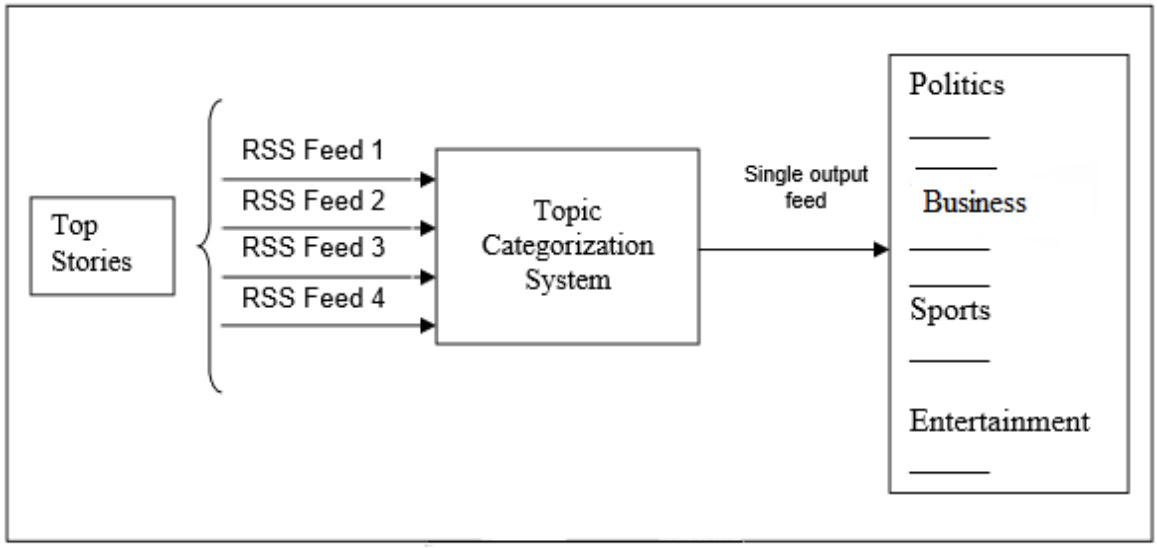


Figure 5-5 Phase 2 Overview

5.2.2 Overview

The categories predefined for this project include Entertainment, Sports, Business and Politics. We have created the ontologies [25] corresponding to each of the categories. These are pre-computed by collected training data for each of the categories from a wide variety of news channels like Reuters, CNN, US News, NPR, Fox News etc.

5.2.3 Design and Implementation

We will explain briefly the detailed algorithm to classify the news feed and get its corresponding category.

Algorithm

The algorithm consists of

- Start by training large numbers of pre-classified documents, during the training phase a database is populated with information about how often certain words appear in each type of document.

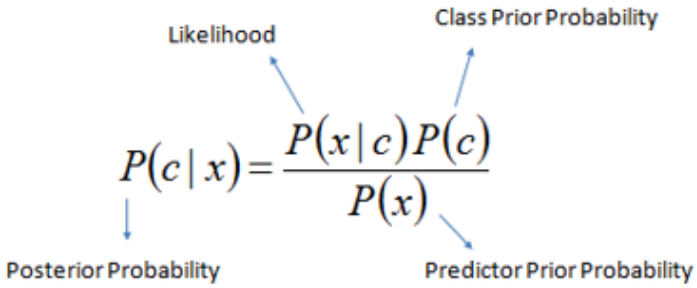
- These documents are then input to the pre-processing algorithm. Following are the steps of this algorithm:
    - Stop words or fluff words are removed from the description.
    - Stemming is carried out on the remaining terms to get the root words.
    - Frequency of every term within the description is computed. (Frequency is the number of times a particular term occurs in one description) Let us denote the collection of terms (content words) in a description by term-list and the corresponding frequency values by the frequency-list.
- A normalized frequency list is computed corresponding to every frequency list using the formula: f1/sqrt (f12 + f12 +……. + fn2), f2/ sqrt (f12 + f12 +……. + fn2), till fn/ sqrt (f12 + f12 +……. + fn2) Where: n = number of content words in one description f1, f2,..., fn = corresponding frequencies of the n content words.In figure below ,small pseudo code for training process .

```
PROCEDURE TRAIN_NAIVE(int N)
For each class c
        Initialize the class feature vector with frequencies for zero terms
        For each document in the class training directory
                Calculate the frequency of each term in the document

                Merge the calculated frequencies into the class feature vector
        Calculate the conditional probability of each term in the class feature
        vector Choose N features from the feature vector to represent the class
END PROCEDURE
```

Figure 5-6 Training process

- Once training is complete, All the parsed descriptions (description is an element of an RSS file that contains a summary of the news story) are retrieved from database then the text must pre-processed as in the previous two step before submitted to the classifier which will return its category. The classifier was used is Naïve Bayesian classifier. Bayes theorem provides a way of calculating the posterior probability, $P(c|x)$, from $P(c)$, $P(x)$, and $P(x|c)$. Naive Bayes classifier assume that the effect of the value of a predictor (x) on a given class (c) is independent of the values of other predictors. The figure below shows the general Bayes equation [26].



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- P(c|x) is the posterior probability of class (target) given predictor (attribute).
- P(c) is the prior probability of class.
- P(x|c) is the likelihood which is the probability of predictor given class.
- P(x) is the prior probability of predictor.

Equation 5-1 General Bayes Equation

The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target as mention before. Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.
The Summery for testing step is shown in the figure.

```
PROCEDURE APPLY_NAIVE(document d)
For a given document d
        For each class c
                Retrieve the conditional probability of each term in d given class c
            Calculate the aggregate probability of all terms of the document Assign
                the document to the class having the highest aggregate probability
END PROCEDURE
```

Figure 5-7 Testing Process

- Finally we need to insert the result in our database, the result of classification will return the category of a specific description then the index of that category will be inserted in the news table that created before under Category_ID attribute.

Tools Used

In this phase we have used Python as programming language as previous phase beside that, we used NLTK the most famous Python Natural Language Processing Toolkit,Python and NLTK enable you to clean up the text by removing punctuation, stopwords and the individual words are then split and normalized into lowercase. Also it provide the following methods that helped in this phase [27] [28]:

- **`nltk.PorterStemmer()`:** Stemmers remove morphological affixes from words, leaving only the word stem.
- **`nltk.corpus.stopwords.words("english")`:** The list of stopwords came from NLTK
- **`nltk.clean_html(example[0]).lower()` :** split and normalized into lowercase.

- **`nltk.sent_tokenize(text)`:** Tokenizers is used to divide strings into lists of substrings. Sentence tokenizer can be used to find the list of sentences.

- **nltk.FreqDist(itertools.chain.from_iterable(processed_texts)):** The ``FreqDist`` class is used to encode "frequency distributions",which count the number of times that each outcome of an experiment occurs.

- **`nltk.NaiveBayesClassifier.train(train_set)`:** applay the navie classifier that we explained before .

5.3 PHASE 3: GEO-TAGGING OF RSS FEEDS

5.3.1 Introduction

Once we have had the article information table in our database updated to the latest news articles, our next job is to extract all the locations mentioned within each article. We aim to build a Geo-tagging system which is the process of adding geographical identification metadata to various media such as a geo-tagged photograph or video, websites, SMS messages, QR Codes or RSS feeds and is a form of geospatial metadata. This data usually consists of latitude and longitude coordinates. 'Geo-tagging' the articles will help in computing a counter for each geographical location on the map, and placing pins indicating the number of news referring to a particular place at a particular hour [29].

5.3.2 Design

Detecting and recognizing geographic locations (toponyms) in news media is a well-established field with many commercial and open source tools available.

Geo-tagging systems typically are made up of two stages. First, text is processed to identify possible place names, and geo-located to create a list of possible physical locations. The technical challenge here is separating place names with names of other entities like people.

The second function of geo-parsing systems is to associate a single latitude and longitude with each location mentioned, called toponym resolution or geographic name disambiguation. These typically employ gazetteer-based approaches, combined with heuristics, to select among candidate locations. The technical challenges of this stage have to do with locating the place mention to the right place [30].

Our work builds on the concept of geographic focus developed for geo-parsing news stories. The implementation of the pipeline outlined:

- Spotting (toponym recognition).
- Disambiguation (toponym resolution).

5.3.3 Implementation

We successfully found CLAVIN which is a software packaged that helped us to build geo-tagging system on the parsed RSS feeds we have already explained in the previous chapter , and In this section, we extensively how we used CLAVIN to achieve our requirements and the results obtained.

The CLAVIN package consists of many libraries and dependencies that are useful to us in this project. We specifically needed `GeoParser`, `GeoParserFactory` and `WorkflowDemo` classes from the package [31].

This phase is implemented using Java, JDK 1.7.0 using Eclipse SDK `4.2.0` as the Java Editor. It uses external stanford-corenlp-3.4.1.jar and clavin-2.0.0-jar-with-dependencies.jar as an external jar files [32].

CLAVIN's process for processing text consists of three distinct phases. In the figure below the general class diagram for this phase, we will start explain briefly the steps until we insert the results of this system in our database as follows:

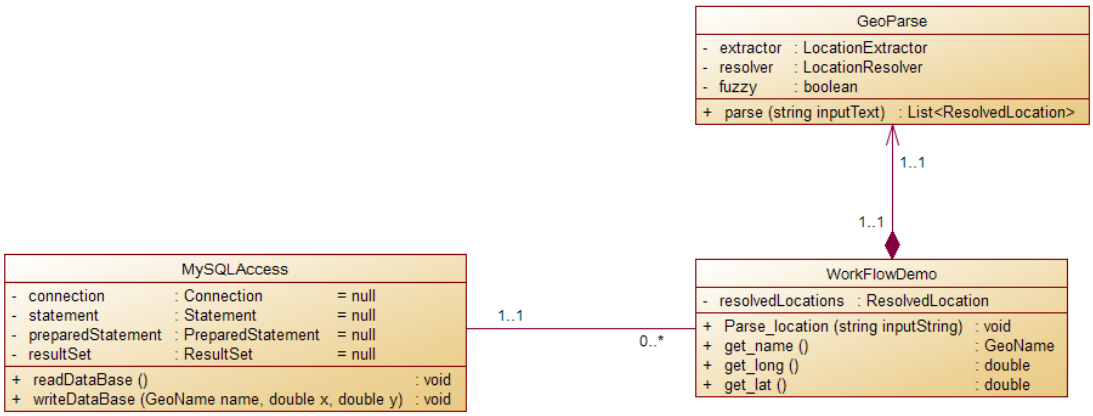


Figure 5-8 Class diagram of the classes and functions used

Gazetteer Model

CLAVIN's worldwide view of geospatial "ground truth" is represented by the gazetteer model. It contains a massive collection of named locations – the cities, countries, mountains, rivers, airports, etc. that you are most likely using CLAVIN to find. It also contains a semantic model for describing these geospatial entities. CLAVIN's gazetteer model is primarily based on the **GeoNames.org** geographical database.

The gazetteer model located in the **com.bericotech.clavin.gazetteer** package contains GeoNames. It is a first-class object in CLAVIN's gazetteer model; represents some named location in the world; consists of the three elements listed above, plus many additional attributes including name, latitude/longitude coordinates, population, elevation, etc

Extraction

CLAVIN uses a `LocationExtractor` to extract location name strings from text, producing a list a `LocationOccurrence` objects

Resolution

Taking the results of the extractor, the `ClavinLocationResolver` resolves each `LocationOccurrence` to the most likely `GeoName` it represents, yielding a `ResolvedLocation`.

The Extraction and Resolution phases can be broken down even further. These are the discrete steps comprising the entire processing workflow:

1. Sets up a connection with the database and reads the title and description from the news table and submits it to CLAVIN.
2. Location names are extracted from the text as `LocationOccurrence` objects by the `LocationExtractor`.
3. For each `LocationOccurrence` object, the location name index is searched for all possible matches.

4. Using the collection of `LocationOccurrence` objects extracted from the text, along with the set of all possible matches for each `LocationOccurrence i`n the location name index, the best `GeoName` object is selected for `LocationOccurrence` by the `ClavinLocationResolver`

5. Resolution results are returned as a list of `ResolvedLocation` objects with `GeoName` for each we interest in GeoID attribute.

6. Insert the ArticleID and its corresponding GoeID in news_Geonames table in our database to match each GeoID with GeoNames database that we have imported into a table in order to get the longitude and latitude for each article.

5.4 PHASE 4: GRAPHICAL USER INTERFACE

When it comes to the front-end division of our project, the spotlight will mainly be on a map. No map entails to no project. Thanks to Google Maps API, this step is not a difficult one. All we have to do is display the correct data at the precise locations brought up by previous steps. Once we've figured out how to finish designing an eye-catching map, all we will have to do is add finishing touches.

5.4.1 Google Maps API [JavaScript]

Figure 5-9 Google Maps API provides an open source map for our website

With Google Maps API, we can easily convert the coordinates of the locations extracted to points on a map. We can assign pins to these points and indicate the number of news of that particular location. Some features that we need might not be present in the latest version of Google Maps API, such as, numbering these pins, and allowing a dialog box to pop up and display the list on news URLs related to a pin linked. For that, we need to work on our own. Google Maps API also gives you the privilege to customize the styles according to your desires. By default, the styles of map look like the image displayed in Figure 1, rather known as the terrain map. In this section, we will get in depth on what we did to make a responsive map that is user friendly.

To embed the map in our website, all we had to do is register a key to monitor our quota. Google Maps API provides its services for free until a specified daily quota, exceeding it will result in payment options. Constructing the map in our website is done by including this class onto our JavaScript `google.maps.Map`. This class extends and MVC object `google.maps.MVCObject`.

5.4.2 Map Construction

First, we need to load the API to our website using:

```
<script src="http://maps.googleapis.com/maps/api/js"></script>
```

Then, we will construct our map class with our script map function that we defined. The following code shows how the map is constructed:

```
var myLatlng = new google.maps.LatLng(0,0);
          var myMap = {
                    zoom: 2,
                    center: myLatlng,
                    mapTypeId: google.maps.MapTypeId.ROADMAP,
                    minZoom: 2,
                         disableDefaultUI: true
                    }
          var               map               =               new
google.maps.Map(document.getElementById("map"), myMap );
```

We defined the center of our map to the coordinates of the center of the globe. To do so, we need to include a `google.maps.LatLng` class. The coordinates of the center of the globe is (0, 0). We also set the zoom to 2, so that all corners of the globe are evident, and no region is hidden. The user cannot zoom out more than that limit. The Map type is a roadmap, since locations extracted may include streets and roads. Finally, we can freely customize our map by disabling the default UI.

5.4.3 Customization

Since we don't need an accurate illustration of the geographical structure of our map, we eliminated the varying shades in the map. For the water, we used a basic and safe blue colour, and we set the land to white with black geographical and road/street outlines. The output looked like this from a distance:

Figure 5-10 The Google Maps API after customization

And by zooming in to a particular location:

Figure 5-11 A zoom in will display the more details of cities, towns and streets

5.4.4 **Numbered Pins**

Placing multiple pins can be done by including the `google.maps.Marker class`.It has includes a long list of various methods, but unfortunately, they do not include changing the number on the pin to note the number of news. For that we need to include an independent custom-made class called `MarkerWithLabel`. This class is constructed within a for-loop. The for-loop is limited to the number distinct locations extracted, in each loop, the `MarkerWithLabel` class is called and a number is put.

The length of the loop and the number of news in each location is determined by a pre-defined SQL query indicated with a php tag:

```
SELECT geonames.geonameid , geonames.longitude,
geonames.latitude,geonames.asciiname, junction.count FROM
geonames INNER JOIN (
SELECT COUNT(*) AS count, temp.geonameid FROM (
  SELECT DISTINCT news_data_geonames.geonameid,
news_data_geonames.ID FROM news_data_geonames)
AS temp Group BY temp.geonameid)
AS junction ON geonames.geonameid = junction.geonameid;
```

The outcome of this query is an array of coordinates (both longitude and latitude), their corresponding IDs in the GeoNames table and the count of news. This array is echoed to the javascript map.js file, which loop through each row to display the pin. This is a screenshot of the final output:

Figure 5-12 After including pins to resemble the number of news in each location

Obviously, the markers are all congested and the numbers are overlapping. Despite the fact that Google Maps API do support various techniques that can be used to display a large number of markers close to each other, it comes at the cost of narrowing the map to a limited region, and is not effective in our entire map. Added to that, the `MarkerWithLabel` class is no part of the Google Maps API package. The reason behind the overlapping numbers is that close pins have similar indexes, and they appear as a layer above all markers, causing visual confusion. The solution would involve putting the label's inside the corresponding pin but this is not possible because Google does not provide an API to return the pin .

At the moment, there is no good workaround. An easy fix would be possible if Google provided a method for returning the used for placing the pin on the map [33].

5.4.5 **Pop-Up Box**

Also included in our scope is the option of clicking a pin to retrieve all the news related to the pin's location. There are numerous ways to do so, but the easiest is by including a jQuery dialog box with custom UI. It can open content in an interactive overlay. Therefore, within our map.js code, we will add a listener for a click action to each pin. As soon as the pin is clicked, the dialog box pops up. Since javascript is on the client-side, and php is on the server-side, the php code is executed initially. A click event can only be passed from a javascript to a php tag with and XML HTTP request. Therefore, we used AJAX's post action, to post the geoname ID of the clicked pin to the php tag. The php tag then executes another SQL query to list all the news related to the clicked pin's location:

```
SELECT DISTINCT news.Link, news.Title,
news_data_geonames.geonameid FROM news
INNER JOIN news_data_geonames ON
news.ID = news_data_geonames.ID AND
news_data_geonames.geonameid=.$var
```

Where '$var' is the geoname ID received from the AJAX post. The output is the array of URL links, article title and the geoname ID of the location. This array is echoed back to the HTML tag so that it can be displayed. The following is the outcome:

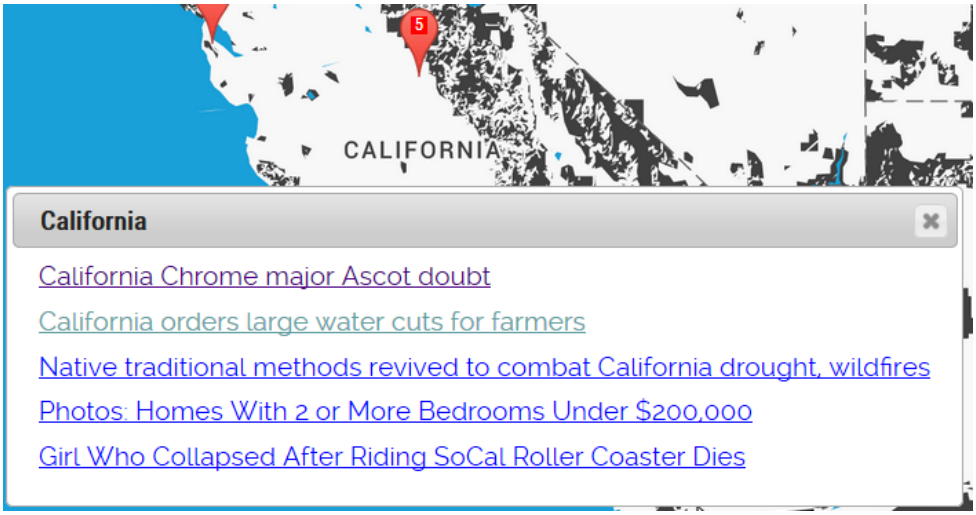


Figure 5-13 List of news present in California State

The dialog box include title of the location clicked, and the list of news hyperlinks. A previously visited link turns purple, and a hover makes the link turn green. These dialog boxes resize automatically according to content, and appear a distance away from the point of click. If the user wishes to escape, they can click on the cross on the top-left corner of the dialog box.

5.5 TEMPLATE [HTML/CSS]

The last step in finishing our GUI is adding a template with minimal navigation. We must offer the option to filter the news according to Category, and it will seem dull if we only placed a map without a template. This will go against our aim to attract users. The template we used was developed by Pixelhint.com *Magnetic*. We only used the side-bar they made to be put next to the map.

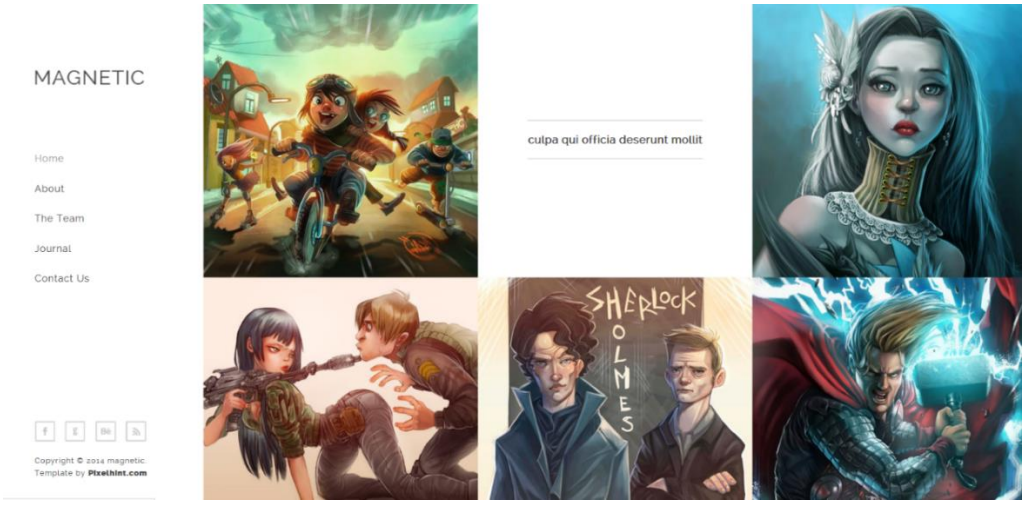


Figure 5-14 Pixelhint's Magnetic theme

We removed all the main contents since we won't be using it, and replaced it with the map we formatted. All we had to do is include our map.js script it in the HTML's <head>, and format the main section tag to accommodate the map. We also replace the links to About, The Team, Journal and Contact Us with Politics, Sports, Entertainment, and Business links. The social links on the bottom of the side-bar were also of no use to us, so we replaced it with the current

date. This date corresponds to the global date and not the one listed on the user's machine. The fonts we used are:

- `Roboto` for location names, dialog box titles, and pin numbers.
- `Raleway` for links and other text.

Finally, we changed the Magnetic logo title to a custom made logo for our project:

Figure 5-15: Project logo

Every time a user wishes to visit our page, and check out the latest news concerning a specific region, this is what he/she will see. A unique, easy-to-navigate interface with a high learnability rate.

Figure 5-16 The final interface

5.6 PHASE 5: HOSTING AND UPDATE

5.6.1 Hosting

We will host our database on regular laptop and we will consider it as our server by applying Port forwarding.

Port forwarding is the behind-the-scenes process of intercepting data traffic headed for a computer's IP/port combination and redirecting it to a different IP and/or port. A program that's running on the destination computer (host) usually causes the redirection, but sometimes it can also be an intermediate hardware component, such as a router, proxy server or firewall.

In short, port forwarding is used to keep unwanted traffic off networks. It allows network administrators to use one IP address for all external communications on the Internet while dedicating multiple servers with different IPs and ports to the task internally. Port forwarding is useful for home network users who may wish to run a Web server or gaming server on one network.

The network administrator can set up a single public IP address on the router to translate requests to the proper server on the internal network. By using only one IP address to accomplish multiple tasks and dropping all traffic that is unrelated to the services provided at the firewall [34].

To start, we need to figure out what our network's default gateway IP address is.We will use this address to access the router's configuration page, and administrative tools.
It is important to remember that each router is different. Most routers will ask for the same information; Service/Application Name, External Port, Internal Port, Protocol and Device IP [35].

- **Service/Application:** The name of the device/service.
- **External Port:** select a single port; for example 8080.
- **Internal Port:** select a single port; for example 8080.
- **Protocol:** Depending on the device this could be either "TCP" or "UDP". If unsure set this as "Both".
- **Device IP:** This is the internal IP address of the device you are connecting to.

5.6.2 Update

In this last phase, there are some points to complete our project and make it applicable to use and performance very well, those points are as follow:

- Automation of system using batch file and task scheduler.
- Check for new feeds every hour.
- Delete old feed within 24 hours from its insertion.

5.6.3 Automation of system using batch file and task scheduler

To run our system automatically without need to run each phase separately by hand we use batch file, we create a batch file using some command for each phase according to the written

programming language then save it , When we Run the batch file it will be runs all the commands in it.

The Command for Each Phase

- Phase 1 & phase 2 <Python platform>:
    - Firstly we need to convert our python scripts into executable file using py2exe module which will produce setup file for each script.
    - Then, we write this command inside the batch file:

```
< Executable_file_path/file_name.exe >
<pytho_program_file_path/file_name.py>
```

- Phase 3 <java platform >
    - The command inside batch file:

```
set path= %CLASSPATH% (path of java jdk)
javac class name.java
java class name
```

- Phase 4 <PHP platform>
    - Add php to your path variable.
    - The command inside batch file : `php path\to\file.php`

Task Scheduler is one of those items hidden in Windows System Tools .It's a relatively simple program that allows you to schedule your scripts and batch files to run on a regular basis.

To make our scripts run automatically, we will use Windows Task Scheduler to create a task that the operating system runs at every hour. The task can point a .bat file (for multiple scripts). Below a figure for windows task schedule [36].

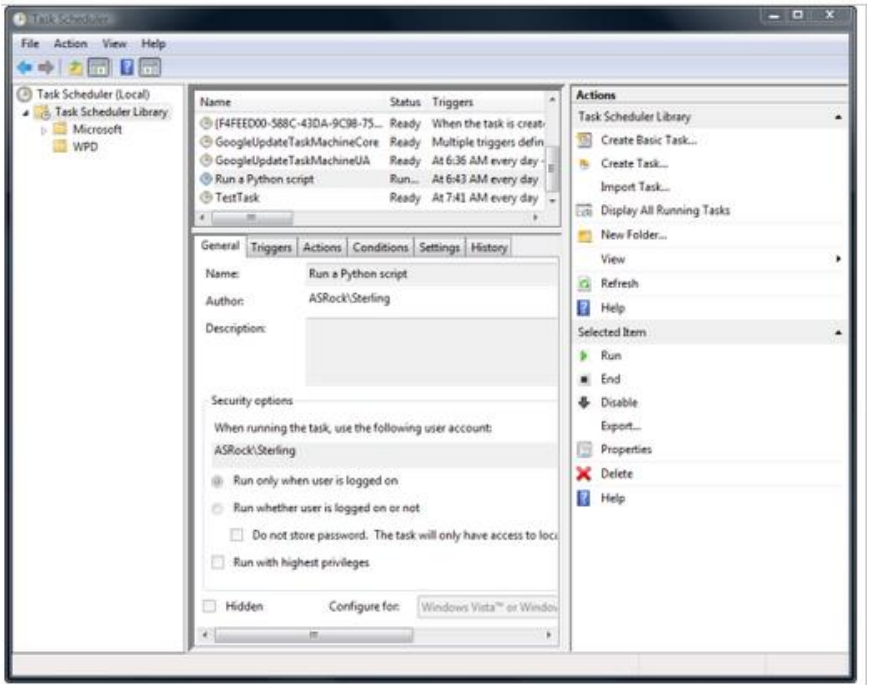


Figure 5-17 The windows task schedule

5.6.4 **Check for update:**

We need to check if there are new feeds to parse every hour in order not to apply category classification and geotagging phases on already existed feeds.

In order to apply that, python provides ETags and Last-Modified headers are two ways that feed publishers can save bandwidth and time [37].

The basic concept is that a feed publisher may provide a special HTTP header, called an ETag, when it publishes a feed. You should send this ETag back to the server on subsequent requests. If the feed has not changed since the last time you requested it, the server will return a special HTTP status code (304) and no feed data.

Inside the batch file we will the return value from parsing phase of the Etag statue ,if the statues code equal 304 ,it will be considered as trigger to the classification and geotagging phases ,otherwise the is no need to run them which will save time. In the case if there are new feeds, we need to memorize the id of last insertion to start from it.

5.6.5 **Delete Old Feeds:**

Our project depends on representing daily news without any need to recording past news ,In order to not increase size of our database ,we apply a small query inside php file that check if the time of any feed since insertion is exceeded 24 hours .this query run automatically every hour within the batch file. Below the query that was executed [38] [39]:

```
SELECT info FROM table
WHERE date > UNIX_TIMESTAMP(NOW() - INTERVAL 24 HOUR);
```

6 TESTING

6.1 UNIT TESTING:

Unit testing is a software development process in which the smallest testable parts called units are individually scrutinized for proper operations. In the unit testing process we will test the four phases of the application which are parsing, categorization, location resolving (CLAVIN) and the front end (GUI).

6.1.1 Parsing Phase

At the first phase of testing we have selected a number of articles on the BBC News website, and checked if our parser has found their links and parsed their attributes into our database.

Input:
BBC's News RSS Feed Link: http://feeds.bbci.co.uk/news/rss.xml?edition=uk

Output:
As shown in the Figure 6-1, below

Link	Title	time	description
http://www.bbc.co.uk/news/uk-33322789#sa-ns_mchann...	Injured Tunisia Britons flown home	2015-06-30 09:17:42	Four British tourists seriously injured in the Tun...
http://www.bbc.co.uk/news/world-europe-33322754#sa...	Tsipras asks Greeks to vote no	2015-06-30 09:17:42	Greek PM Alexis Tsipras urges voters to reject cre...
http://www.bbc.co.uk/news/uk-33315691#sa-ns_mchann...	Counter-terrorism exercise in London	2015-06-30 09:17:42	Police officers, soldiers, emergency services and ...
http://www.bbc.co.uk/news/education-33310736#sa-ns...	'Coasting schools' face tough targets	2015-06-30 09:17:42	Hundreds of schools are being told to raise their ...
http://www.bbc.co.uk/news/world-asia-33323419#sa-n...	Military plane crashes in Indonesia	2015-06-30 09:17:42	At least five people are killed as a military tran...

Figure 6-1 The output of feed parsing phase

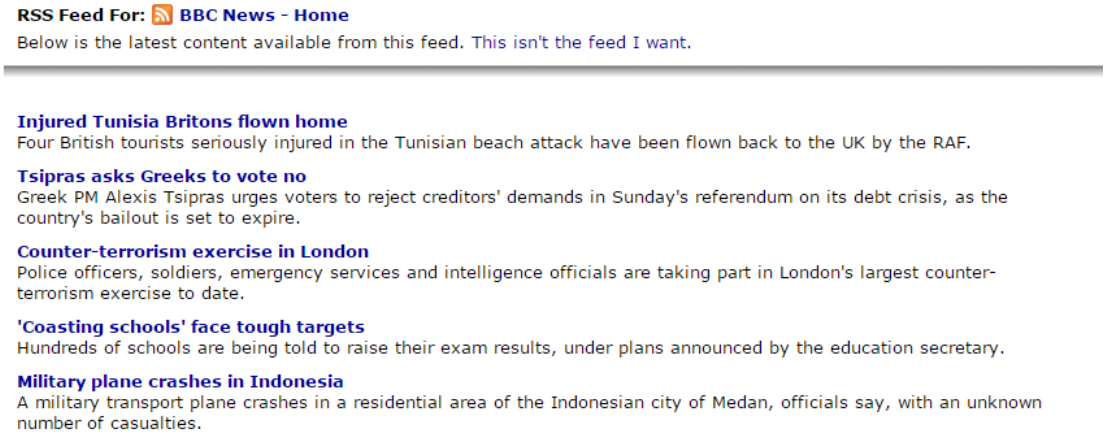


Figure 6-2 The RSS feed of BBC News

6.1.2 Categorization Testing Phase

At this phase we use naive Bayesian classifier, where we provided it with the description as an input, the expected output will be the category type of the article, whether it be political, sports, entertainment or business.

Input:

'"The opening night of William Tell at the Royal Opera House is marked by boos over a nude rape scene"

Output:

```
In [11]: %run "C:\Users\ghada\Canopy\scripts\classify.py"
<class '__main__.Classifier'>
['The opening night of William Tell at the Royal Opera House is marked by boos over a nude rape scene.'] -> classified as: Entertainment
```

Figure 6-3 The output of categoriation phase

Figure 6-4 The actual category of input feed from the BBC news

6.1.3 CLAVIN Phase

At this phase we have tested the CLAVIN tool by providing it with an article's title and description, and expected the output to be the longitude and latitude of the places mentioned in the provided information.

Input:

"Injured Tunisia Britons flown home
Four British tourists seriously injured in the Tunisian beach attack have been flown back to the UK by the RAF"

Output:

From the CLAVIN web application:

Name	Lat, Lon
Tunisian Republic	34, 9
United Kingdom of Great Britain and Northern Ireland	54.75844, -2.69531

Figure 6-5 The output of CLAVIN phase

6.1.4 GUI testing phase

At this phase we use the longitude, latitude and name deduced from the CLAVIN tool as an input and check on the Google Map whether this locations has a marker over this region. Labelled on the marker is the number of articles covering news about this location, and the list of links shown on clicking on the marker include the original article from the source.

Input:

```
var myCenter=new google.maps.LatLng(51.508742,-0.120850);
```

Output:

The Output is shown below

Figure 6-6 The output of GUI phase

6.1.5 Update Testing

We know that on average, the feed parser takes less than a minute to parse around 200 articles and that the geo-tagger takes half a second to resolve locations in each article. What we don't know is what could be the maximum total time to insert an article into our database, and view it on the map. We left the entire system to update on an hourly basis and logged the total time taken for a total of 12 hours. This was the result:

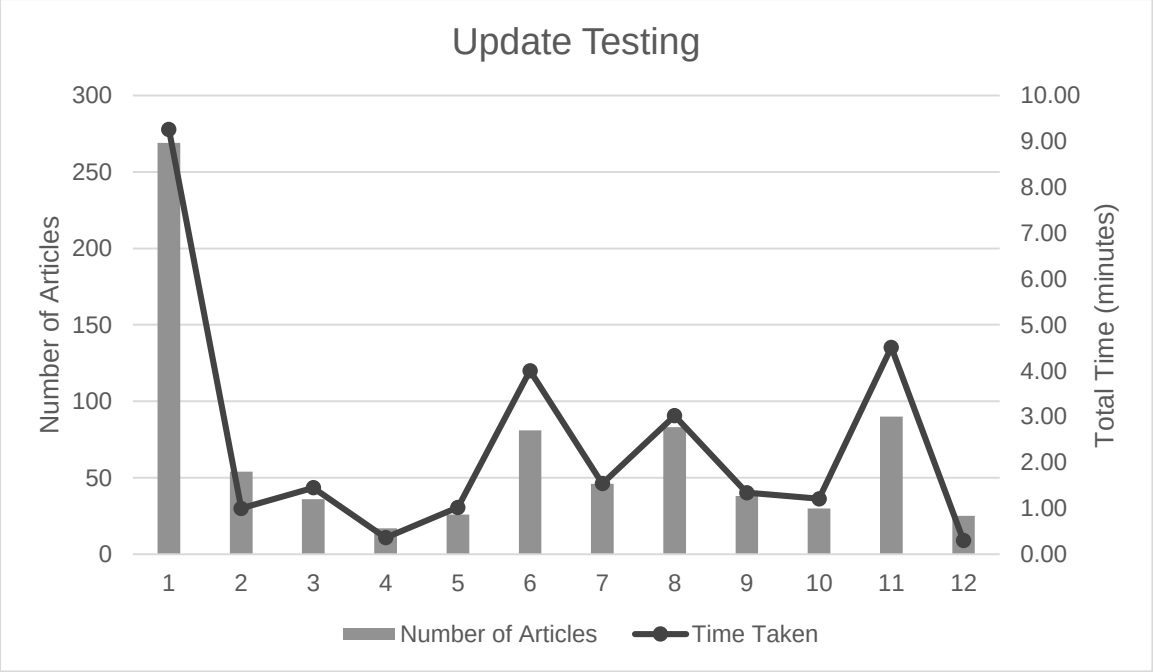


Figure 6-7 The Result of an Hourly Update on the Span of 12 hours

At the start of the chart, the system is on start-up and receives the first batch at the start of the day. The amount of input articles decrease on the second run. This is due to the fact that it does not parse articles that are already in the database, and does not geo-tag and classify it an old article. All in all, the chart indicates that time taken to for the database to update itself relies greatly on the number of news articles, the more the news articles, the greater time.

6.1.6 Stress Testing

It is difficult to control the number of articles getting into the system since it is entirely dependent on the news provider. It may be impossible to stress test on the feed parser, but it is possible for the rest of the system. In this case, we forced the remaining phases (phase 1 excluded) to run on an entire day's data, 945 articles in total. Earlier, we only ran through new data and not a day's work. The result took all in all 1 hour and 14 minutes.

6.2 INTEGRATION TESTING

We guaranteed the output for each phase is as initially expected. We now wish to see the output once we integrate the entire system. The best way to test our system in its entirety, is by checking if a news headline on one of the news sources is present on our map at the correct location. Take for instance the following article issued by the BBC World News:

Figure 6-8 A BBC News headline concerning Cuba

It is apparent that the country of topic is Cuba. We expect this headline to appear when we locate Cuba on the map and list all the news headline. We also expect that clicking on that headline URL will open a new tab to this particular article.

Heading back to our map, and zooming in on Cuba, we can see that there are several news, one of them is "Cuba stamps out mother-to-child HIV". This successfully directs us to the original source of the article:

Figure 6-9 Successful Output

7 CONCLUSION

It has been shown through this project that we can classify news further more than continental regions, and sharpen the accuracy up to cities, towns and street names. We were capable of revolutionizing how news can be viewed. This project can be commercially distributed across the World Wide Web. Users can access it from any place in the World, and get informed of the latest news inside and outside their region. We are glad to achieve all this in minimal time using a simple PC. The projects development can be improved to make space for more user access, greater data storage and more accurate results.

In the end, we were capable of implementing a news aggregator web application with more classification features of news articles and an attractive user interface. The 5 main phases worked as expected and the project was smoothly executed. We gathered news articles from several trusted news providing websites by parsing their corresponding RSS Feeds. Data of significance of each article was then stored. A geo-tagger, CLAVIN, resolved the list of locations mentioned within the article, geographical information related to the article was stored. Articles were further classified according to defined categories. All the data stored was queried to visualize the capacity of news in each location. The website that acted as a user interface was hosted and remotely accessed. Data was updated with every passing hour in every day.

We used our knowledge of OOP concepts, machine learning and pattern recognition, parallel processing, software design, and database management and web development to make this project a success. That being said, this project has also widen our horizons and gave us an insight on other things in the learning process. The most significant of them was integrating various tools and frameworks to complete an entire system.

7.1 FUTURE WORK

As a team, we wish to do more using additional resources to enhance this project. We are determined that this project can be a commercial success in the future and not just an undergraduate senior project. Here are some of the things we could add:

- Add more news sources to our list.
- Add more categories and enlarge our training set.
- Create a more dynamic interface that shows pulses over the region that captures an event.
- Allow variable size of pulses according to the importance and news trend of the location being shown.
- Link the application with social media feeds, such as Twitter and Facebook, including discussions about the posted news.
- Summarize all headlines of a particular story into one headline.
- Start to introduce user accounts, where the application can provide the user with their favourite type of news deduced from the user's frequency of checking the news.
- Use tools that can extract news from wider range of media outlets and different languages.
- Language detection to relate news published in different languages for the same event.

- Include an Android Mobile Application to the package.
- Edit the GUI for a more appealing website and more interactive interface.
- Hosting our web application on a server.
- Add more data concerning categories and life events to address more locations. (For example, link Premier League to UK, Hollywood to USA etc.)
- Reserve a bigger database for more data storage, and faster data retrieval and insertion.

8 REFERENCES

[1] P. R. C. f. t. P. a. t. Press, “How Often People Watch Local TV News,” 13 March 2005.

[2] R. Dobelli, “News is bad for you – and giving up reading it will make you happier,” 12 April 2013.

[3] P. R. C. f. t. P. a. t. Press, “In Changing News Landscape, Even Television is Vulnerable,” 27 September 2012.

[4] P. R. C. f. t. P. a. t. Press, “Watching, Reading and Listening to the News,” 27 September 2012.

[5] P. R. C. f. t. P. a. t. Press, “Interest in Foreign News Declines,” 6 June 2012.

[6] “BBC News Most Popular Now,” BBC World News, 2015. [Online]. Available: http://news.bbc.co.uk/2/shared/bsp/hi/live_stats/html/map.stm. [Accessed November 2014].

[7] “Techopedia,” 2010. [Online]. Available: http://www.techopedia.com/definition/86/geotagging. [Accessed November 2014].

[8] A. DuVander, “Twitter's Many Geo APIs,” ProgrammableWeb, 26 April 2010. [Online]. Available: http://www.programmableweb.com/news/twitters-many-geo-apis/2010/04/26. [Accessed November 2014].

[9] F. Podlaha, “Twitter Tip: Geo-tagging. What is it, how to do it, and for God’s sake, “Why?”,” Businesses Grow, 19 January 2010. [Online]. Available: http://www.businessesgrow.com/2010/01/19/twitter-tip-geo-tagging-what-is-it-how-to-do-it-and-for-gods-sake-why/. [Accessed November 2014].

[10] “Google News,” Google, 2013. [Online]. Available: https://news.google.com/. [Accessed November 2014].

[11] S. Machlis, “Inside the Google News algorithm,” ComputerWorld, 5 October 2009. [Online]. Available: http://www.computerworld.com/article/2467854/e-commerce/inside-the-google-news-algorithm.html. [Accessed November 2014].

[12] “The GDELT Project,” GDELT, 2013. [Online]. [Accessed November 2014].

[13] T. B. Group, “American newspapers and the internet: Threat or opportunity?,” The Bivings Group, 2007.

[14] T. N. a. T. T. Hao Han, “An Automatic Web News Article Contents Extraction System Based on RSS,” *Journal of Web Engineering*, vol. 8, no. 3, pp. 268 - 284, 2009.

[15] “CLAVIN,” Berico Technologies, 2012. [Online]. Available: https://clavin.bericotechnologies.com/. [Accessed 2015].

[16] R. Cirrius, “Introducing "CLAVIN" (Cartographic Location And Vicinity INdexer),” 30 September 2012. [Online]. Available: http://www.gettingcirrius.com/2012/09/introducing-clavin-cartographic.html. [Accessed 2015].

[17] B. Technologies, “CLAVIN Capabilities Overview,” Berico Technologies, Reston, VA, 2012.

[18] S. D. S. Mark Graham, “Information Graphics, Mapping the GeoNames Gazetteer,” 2010. [Online]. [Accessed 2015].

[19] “RSS Tutorial for Content Publishers and Webmasters,” [Online]. Available: https://www.mnot.net/rss/tutorial/. [Accessed 1 7 2015].

[20] “The Anatomy of an RSS Feed,” [Online]. Available: http://www.webreference.com/authoring/languages/xml/rss/feeds/index.html. [Accessed 1 7 2015].

[21] “RSS - Really Simple Syndication,” [Online]. Available: http://www.xul.fr/en-xml-rss.html. [Accessed 1 7 2015].

[22] “feedparser 5.2.0 documentation,” [Online]. Available: https://pythonhosted.org/feedparser/. [Accessed 1 7 2015].

[23] “Python RSS Code,” [Online]. Available: https://wiki.python.org/moin/RssLibraries. [Accessed 1 7 2015].

[24] P. A. P. G. S. J. S. A. Bhushan Pendharkar, “Topic Categorization of RSS News Feeds,” 2007.

[25] “Categories dataset,” [Online]. Available: https://wiki.csc.calpoly.edu/481-W09-CagetorizeRSSNN/browser/RSS/trainingData. [Accessed 1 7 2015].

[26] “Naive Bayesian,” [Online]. Available: http://www.saedsayad.com/naive_bayesian.htm. [Accessed 1 7 2015].

[27] “Explore Python, machine learning, and the NLTK library,” [Online]. Available: http://www.ibm.com/developerworks/library/os-pythonnltk/#list7. [Accessed 1 7 2015].

[28] “Welcome to NLTK-Trainer’s documentation,” [Online]. Available: http://nltk-trainer.readthedocs.org/en/latest/index.html. [Accessed 1 7 2015].

[29] “Geoparsing,” [Online]. Available: https://en.wikipedia.org/wiki/Geoparsing. [Accessed 1 7 2015].

[30] R. B. E. Z. Catherine D’Ignazio, “CLIFF-CLAVIN: Determining Geographic Focus for News Articles,” p. 5, 2014.

[31] “CLAVIN,” [Online]. Available: https://clavin.bericotechnologies.com/clavin-core/. [Accessed 1 7 2015].

[32] “Berico-Technologies/CLAVIN-NERD,” [Online]. Available: https://github.com/Berico-Technologies/CLAVIN-NERD. [Accessed 1 7 2015].

[33] “Issue 24, MarkerWithLabel: problem of overlapping labels when two markers are in the same position,” Google Maps. [Online]. [Accessed 2015].

[34] “How To Forward Ports on Your Router,” [Online]. Available: http://www.howtogeek.com/66214/how-to-forward-ports-on-your-router/. [Accessed 1 7 2015].

[35] “General Port Forwarding Guide,” [Online]. Available: http://www.noip.com/support/knowledgebase/general-port-forwarding-guide/. [Accessed 1 7 2015].

[36] “How to schedule a Batch File to run automatically in Windows 7 | 8,” [Online]. Available: http://www.thewindowsclub.com/how-to-schedule-batch-file-run-automatically-windows-7. [Accessed 1 7 2015].

[37] “ETag and Last-Modified Headers,” [Online]. Available: https://pythonhosted.org/feedparser/http-etag.html. [Accessed 1 7 2015].

[38] “Delete Record After 24 Hours How To,” [Online]. Available: http://bytes.com/topic/mysql/answers/864526-delete-record-after-24-hours-how. [Accessed 1 7 2015].

[39] “QUERY RECORDS WITHIN 48 hours,” [Online]. Available: http://dba.stackexchange.com/questions/55460/query-records-within-48-hours. [Accessed 1 7 2015].

[40] “Python RSS Code,” [Online]. Available: https://wiki.python.org/moin/RssLibraries. [Accessed 1 7 2015].

[41] “https://wiki.python.org/moin/RssLibraries,” [Online]. Available: https://wiki.python.org/moin/RssLibraries. [Accessed 1 7 2015].

9 APPENDIX A: ACKNOWLEDGEMENT

The completion of this work could not have been possible without the participation and assistance of many people whose names may not be all enumerated. Their contributions are sincerely appreciated and gratefully acknowledged. However, we would like to express our deep appreciation and indebtedness to our Supervisor Professor Yousry Taha for his efforts and support during the past year throughout all stages of development of this project. We also would like to thank our parents, to whom we are indebted with every accomplishment we have and will ever achieve in life, for their endless support and encouragement all along our way. We also would like to thank our friends and family for their kind cooperation and moral support which helped us complete this project.

We would like to show our gratitude towards all our teachers and Professors who taught us throughout our academic life who have contributed to what we have accomplished today by teaching us all the knowledge we have.

And above All, utmost appreciation to the Almighty God, the author of knowledge and wisdom, for His countless blessings and mercy upon us.

10 APPENDIX B: COPYRIGHT FORM

Project Title: The Global Daily Journal

We the team:

ID	Member Name	Member Signature
1308	Alaa Ahmed Nagaty	
1321	Reem Nasser Ezz Eldein	
1337	Ghada Ibraheem Fahmy	
1369	Yara Fareed Fahmy	
1378	MennatAllah Ahmed Ward	

Hereby assign our copyright of this report and of the corresponding executive summary to the Computer and Communication Engineering Department (CCP) of Alexandria University. We also hereby agree that the report or demo from our oral presentations is becoming full property of the CCP Department.

Publication from this report does not constitute approval by Alexandria University, the CCP Department or its faculty members of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.