TRAIL TRACK

DATA: RETRIVER, PROCESSOR AND GENERATOR

Enrollment No : 12103449

Name : Anshit Agarwal

Enrollment No : 12103453

Name : Karan Dhingra

Supervisor : Dr. Manish K Thakur

Submitted in partial fulfilment of the degree of

Bachelor of Technology

in

Computer Science Engineering

Department of Computer Science Engineering & Information Technology

Jaypee Institute of Information Technology

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled "TRAIL TRACK DATA: RETRIVER, PROCESSOR AND GENERATOR" submitted by me to "Jaypee Institute of Information Technology", Noida in partial fulfillment for the requirement for the award of degree of B.TECH in COMPUTER SCIENCE AND TECHNOLGY DEPARTMENT is a record of bonafide project work carried out by me under the guidance of Dr. Mr. Manish K Thakur. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other diploma or degree in this institute or any other institute or university.

Place: Noida Signature of Candidates

Date: 23 December, 2015 Anshit Agarwal Karan Dhingra

CERTIFICATE

This is to certify that the project report entitled "TRAIL TRACK DATA: RETRIVER, PROCESSOR AND GENERATOR" submitted by Anshit Agarwal (12103449) and Karan Dhingra (12103453), in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology (B. Tech) of Jaypee Institute of Information Technology University, Noida has been carried out under my supervision. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of Supervisor

Dr. Mr. Manish K Thakur

Asst. Professor (Sr. Grade)
Department of CSE/IT
Jaypee Institute of Information Technology, Noida, India

23 December, 2015

ACKNOWLEDGEMENT

We take pride to express our deep sense of gratitude to Dr.Mr. Manish K Thakur, Assistant Professor, Department of Computer Science Engineering & Information Technology, JIIT Noida; for his undeterred guidance, continuous encouragement, and support. Moreover, I would also take this opportunity to thank Dr. Prakash Kumar & Dr. Shradha Porwal, external guides, for sparing their valuable time and keeping us on the right track. This project would not have been possible without any of the above mentors.

Further, I take this opportunity to whole-heartedly thank Prof. Dr. Sanjay Goel, Professor and Head (Dept. of CSE/IT) & Program Chair and all teaching staff and members working as limbs of our university for their non-self-centered enthusiasm coupled with timely encouragements showered on us with zeal, which prompted the acquirement of the requisite knowledge to finalize our course study successfully. We would like to thank our parents for their support.

It is indeed a pleasure to thank our friends who persuaded and encouraged us to take up and complete this task. And last but not least, we express our gratitude and appreciation to all those who have helped us directly or indirectly toward the successful completion of this project.

Place : Noida Signature of the Students

Date : 23 December, 2015 Anshit Agarwal Karan Dhingra

SUMMARY

The application that we have developed helps corporations identify the risks of investing in individuals by retrieving all the relevant information from the Internet. It helps the administrator crawl data in real time and analyse the results to produce intelligent data. The first part of the project inputs the pre-provided details of the person. The information is then crawled from the trusted sources and added to the database. The crawled information is then processed to produce meaningful and human understandable format. This part uses natural language understanding and generation. The infobank sources, social media sources and the news sources are displayed accordingly under tabs. The final report is then displayed in accordance with a pre defined template.

Table of Contents

1. Introduction

1.1 General Introduction

Risk Analysis and Mitigation were terms unheard before 2005.

Risk Analysis:

Qualitative risk analysis is a project management technique. It is concerned with searching for the possibility and probability of a risk event occurring. Additionally, it helps foresee the impact the risk will have if it does occur. All risks have both probability and impact. Probability is the percentage that a risk event will occur, and impact is the quantity of the consequences of the risk event. Impact typically affects the following elements:

1. Schedule
2. Budget
3. Resources
4. Deliverables
5. Costs
6. Quality
7. Scope
8. Performance

Risk Mitigation:

Risk mitigation is the reduction to the extent of exposure to a risk. Also, it helps reduce the likelihood of its occurrence. Risk management identifies a risk with 100% probability of occurring. But, it is ignored in the company due to a lack of identification and management ability.

Relationship risk appears when ineffective collaboration occurs. These risks directly and indirectly reduce the productivity of the following:

1. Knowledge workers
2. Decrease cost-effectiveness
3. Profitability
4. Service
5. Quality

6. Reputation
7. Brand value
8. Earnings quality

Intangible risk management allows risk management to create immediate value from identification and reduction of risks that reduce productivity.

The application that we have created will help corporations identify the risks of investing in individuals. The information is retrieved from trusted sources on the internet. The information is later processed to produce a concise report.

Web Crawler:

A Web crawler is an Internet robot which browses the World Wide Web. It uses a stack to read the links and text present on a webpage and stores them locally. Later, the stack of links is popped to open the next link and crawl it.

Web search engines and some other sites use Web crawling or spidering software to update their web content or indexes of others sites' web content. Web crawlers can copy all the pages they visit for later processing by a search engine which indexes the pages efficiently.

Crawlers can validate hyperlinks and HTML code. They can also be used for web scraping.

Natural Language Understanding:

Natural language understanding is a subtopic of natural language processing. It is a subfield of artificial intelligence that deals with machine reading comprehension.

It helps with disassembling and parsing input. And is more complex than the reverse process of assembling output in natural language generation because of the occurrence of unknown and unexpected features in the input and the need to determine the appropriate syntactic and semantic schemes to apply to it, factors which are pre-determined when outputting language.

There is considerable commercial interest in the field because of its application to news-gathering, text categorization, voice-activation, archiving, and large-scale content-analysis.

Natural Language Generation:

Natural Language Generation (NLG) is the natural language processing task of generating natural language from a machine representation system such as a knowledge base or a logical form. It could be said an NLG system is like a translator that converts a computer based representation into a natural language representation. However, the methods to produce the final language are different from those of a compiler due to the inherent expressivity of natural languages.

1.2 Critical Analysis:

With the analysis of web crawling there are several issues which have not been discussed. Data is crawled from any webpage by finding data between html tags, each website has different format, and usable data may be available in different tags. Chinese whisper algorithm has been defined for inter-related mathematical data; we needed to implement it for synonyms and similar sentences. For this we need a list of synonyms that will be required to be crawled in real time. Listing synonyms from the internet can result in erroneous data since a single word might have multiple meanings.

Further, here is a list of issues that have not yet been raised with data crawling and processing.

1) Understanding a language is based on important tags. The tags database can become a large and unmanageable file due to synonyms and similar sounding words.

2) In phonetic languages like English, a word has multiple meanings, making it difficult to understand the context of a particular context.

3) The data available on the internet is segregated and might not be true. This creates a problem in looking for data from un-trusted sources.

4) Every website uses a different set of tags to list data.

5) Some websites like Google and LinkedIn cannot be crawled. They need to be textised and scraped.

1.3 Problem Statement:

The aim is to develop a web application that will be used to mine data related to a particular person. The information will be retrieved from trusted sources on the internet. The information will then be processed to produce a concise report.

The project is divided into four parts:

1. The first part of the project will input the pre provided details of the person. As soon as the submit button is pressed, the crawler will start looking for information from the trusted sources.

2. The information will be crawled from the trusted sources and added to the database. The data will be cleaned with the help of a cleaner. The stop words will be removed to rank the importance of the sentences. Each sentence is then searched for the predefined tags.

3. The crawled information will be processed to produce meaningful and human understandable format. This part uses natural language processing techniques to read and enlist important data.

4. The final report will be displayed in the form of a website. The website will be divided into tabs. Each tab contains the specific traits related to a person which will help us learn about the individual and the merits and demerits of investing in him. The current news about popular individuals will be displayed in a different tab.

1.4 Proposed solution and novelty of solution

1.4.1 Proposed Solution:

Crawling all the information about a person is virtually impossible. It is also virtually impossible for a company to read about an individual every time while investing in them. The application provides for easy and fast means extract relevant information about any individual.

Hence, the application crawls data only from trusted sources like Wikipedia and LinkedIn. Further, news websites are also crawled for the information. This information is cleaned of any irrelevant data using a data cleaner. The cleaned data is reduced by removing "Stop-Words" and now the importance of the sentence is measured. The sentences are also divided under tags to help them segregate n the basis of information later during the final report.

The Natural Language Processing techniques like word tagging and text summarization are used to understand the importance of sentences. The final report is displayed in the form of a website divided into tabs containing:

1. InfoBank Sources
2. Social Media Sources
3. News Websites

1.4.2 Novelty of Solution:

This application brings a valuable method to understand and summarize text related to a particular individual. The following list contains a few novelties that our solutions attained.

1) To crawl unknown websites, we are textising all the existing data on the website and then reading the textised form. This data is then cleaned of irrelevant data. Further only important and significant information is saved in the database.
2) Websites like LinkedIn do not allow their data to be crawled. To read and save this data, we are using an OpenSource tool named scrappy. This data is then cleansed of all unwanted noise.
3) The information that is required from the data is retrieved on the basis of tags. But the tags in English language have multiple synonyms and the list of tags can increase over time. To reduce this error, we have made a dynamically changeable tag file.

4) To help the speed of the final website, the previously crawled data is also used to make new inferences. This solves another purpose of scalability.

2. Background Study

2.1 Literature Survey

2.1.1 Summary of Papers

2.1.1 Title: Natural Language Processing in Web Data Mining

Authors: Yue Chen

Publishing Details: School of Computer Science and Technology, Beijing Institute of Technology Beijing, China

Summary: This paper describes the research about Web data mining using Natural Language Processing. System accepts arbitrary data as input from Web document and then extracts information from the document. A new method to implement Web data mining is proposed in this paper. There are three steps in this system. First, the Web document will be decomposed to paragraph, sentence and phrase level. Second, extract information from all sentences. Finally, add the information to the knowledge model. The methods used have proved to be efficient for Web data mining with the experimental corpus.

Web Link: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5607419&tag=1

2.1.2 Title: Graph-Based Methods for Natural Language Processing and Understanding - A Survey and Analysis

Authors: Michael T. Mills and Nikolaos G. Bourbakis

Year of Publication: 2014

Publishing Details: IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS

Summary: This survey and analysis presents the functional components, performance, and maturity of graph-based methods for natural language processing and natural language understanding and their potential for mature products. Resulting capabilities from the methods surveyed include summarization, text entailment, redundancy reduction, similarity measure, word sense induction and disambiguation, semantic relatedness, labelling (e.g., word sense), and novelty detection. Estimated scores for accuracy, coverage, scalability, and performance are derived from each method. This survey and analysis, with tables and bar graphs, offers a

unique abstraction of functional components and levels of maturity from this collection of graph-based methodologies.

Web Link: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6576885

2.1.3 Title: Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems

Authors: Chris Biemann

Publishing Details: University of Leipzig, NLP Department

Summary: We introduce Chinese Whispers, a randomized graph-clustering algorithm, which is time-linear in the number of edges. After a detailed definition of the algorithm and a discussion of its strengths and weaknesses, the performance of Chinese Whispers is measured on Natural Language Processing (NLP) problems as diverse as language separation, acquisition of syntactic word classes and word sense disambiguation. At this, the fact is employed that the small-world property holds for many graphs in NLP.

Web Link:
http://wortschatz.uni-leipzig.de/~cbiemann/pub/2006/BiemannTextGraph06.pdf

2.1.4 Title: Extraction based summarization using a shortest path algorithm

Authors: Jonas Sjobergh and Kenji Araki

Publishing Details: Proceedings of the 12th Annual Natural Language Processing Conference

Summary: An extraction based method for automatic summarization. It is based on finding the shortest path from the first sentence to the last sentence in a graph representing the original text. Nodes represent sentences and edges represent similarity between sentences. Simple word overlap is used for similarity. Traditional sentence weights are also used, making edges to important sentences cheaper.

Web Link: http://www.dr-hato.se/research/shortpath.pdf

2.1.2 Tabular Comparison

Problem	Input	Output	Technique	Tool	Sample
Data mining and sentence recognition.	Articles.	A graph to store the knowledge generated by sentence analysis.	Web data mining	Tag creation and graph generation.	Wikipedia, LinkedIn
Natural language processing and understanding.	Text paragraph, documents.	Graphs: clustering, similarity measure.	Summarisation, text entailment, similarity measure, word sense induction word sense disambiguation.	Graph generation.	The cleansed and ranked reports generated from the crawled data.
Graph-clustering.	Text document.	Clustered graph.	Chinese whisper.	Graph generation.	
Extraction based text summarization	Text paragraphs.	Text summary.	Summarization using shortest path.	Graph generation.	The final data displayed on the website.

2.2 Field Survey and Experimental Studies

2.2.1 Field Survey:

We started reading about the importance of Risk analysis and mitigation a few months ago. This topic has become more relevant than ever after the 2008 financial crisis. To predict the failure of investment beforehand, all the major banking and finance companies in the world employee 100s of people to analyse risks. Risk management and prediction is used widely to analyse the risk before investing in an individuals. To automate the process of risk analysis, we needed to create a tool that can gather data and present it in a concise human readable form.

In Web data mining the initial work is to decide what information will be extracted from the web pages. Knowledge model is designed to describe information that is extracted from the web document. Knowledge nodes are defined to extract information from sentences which contain those nodes.

Chinese whisper is a graph-clustering algorithm used in natural language processing problems such as language separation, acquisition of syntactic word classes and word sense disambiguation.

2.2.2 Experimental Studies:

Risk analysis is a technique used to identify and assess factors that may jeopardize the success of a project or achieving a goal.

This technique also helps to define preventive measures to reduce the probability of these factors from occurring and identify countermeasures to successfully deal with these constraints when they develop to avert possible negative effects on the competitiveness of the company.

Risk analysis can be qualitative or quantitative. Qualitative risk analysis uses words or colours to identify and evaluate risks or presents a written description of the risk, and quantitative risk analysis (QRA) calculates numerical probabilities over the possible consequences.

Quantitative Risk Analysis:

QRA seeks to numerically assess probabilities for the potential consequences of risk, and is often called probabilistic risk analysis or probabilistic risk assessment (PRA). The analysis often seeks to describe the consequences in numerical units such as dollars, time, or lives lost. PRA often seeks to answer three questions:

1. What can happen? (i.e., what can go wrong?)

2. How likely is it that it will happen?

3. If it does happen, what are the consequences?

Pseudo-quantitative risk assessment:

Pseudo-quantitative risk assessments generally assign numbers to the likelihood and consequences for a risk but do not build a mathematical model of the risk as suggested by PRA. The most popular pseudo-quantitative method is probably the risk matrix, which classifies the likelihood of a risk in one category and the consequences in another category. The combination of the likelihood and consequence categories corresponds to a risk level, usually a colour such as red, orange, yellow, and green. A risk matrix is sometimes called a pseudo-quantitative method because the categories may be determined from numbers.

The Risk Advisory Model that most Multinationals use today:

Multinationals like the Big 4 and Major Banks hire people to foresee the security of returns on investment. The main job of the people is to search for the giving data using popular search engines as well as shared company databases to check the financial history of the client. This information is then analysed by a team of Financial Analysts to make intelligent decisions.

Automatic summarization is the process of reducing a text document with a computer program in order to create a summary that retains the most important points of the original document.

Types of Summarization:

The different types of automatic summarization include extraction-based, abstraction-based, maximum entropy-based, and aided summarization.

1. Extraction-based summarization:

In this summarization task, the automatic system extracts objects from the entire collection, without modifying the objects themselves. Examples of this include keyphrase extraction, where the goal is to select individual words or phrases to "tag" a document, and document summarization, where the goal is to select whole sentences (without modifying them) to create a short paragraph summary. Similarly, in image collection summarization, the system extracts images from the collection without modifying the images themselves.

2. Abstraction-based summarization:

Extraction techniques merely copy the information deemed most important by the system to the summary (for example, key clauses, sentences or paragraphs), while abstraction involves paraphrasing sections of the source document. In general, abstraction can condense a text more strongly than extraction, but the programs that can do this are harder to develop as they require the use of natural language generation technology, which itself is a growing field.

3. Aided summarization:

Machine learning techniques from closely related fields such as information retrieval or text mining have been successfully adapted to help automatic summarization.

3. Analysis, Design, Modelling:

3.1 Requirements:

In software engineering (and systems engineering), a functional requirement defines a function of a system and its components. A function is described as a set of inputs, the behaviour, and outputs .Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. The plan for implementing is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture.

Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of "system shall do <requirement>", an individual action of part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model.

3.2 Functional and Non-functional Requirements:

Functional Requirements:

- The user interface is minimalistic and easy to use. It only contains of two search boxes where the user can enter the name and available information.

- The data is cleaned and the information is segregated multiple times before displaying it to the user. Reports are cleaned using stop words and frequent tags.

- Data is crawled even from websites that restrict data crawling.

- A list of news websites will be displayed as the crawling works and the user can read about the source of the information.

Non-Functional Requirements:

- The crawling speed needs to be fast since a lot of data regarding a few people is present on the internet.

- The crawled data needs to be error free. This is achieved by multiple layers of data crawling and cleansing.

- The Natural Language Processing must be quick and error free. This is achieved by tagging the sentences using synonyms and text summarization techniques.

- The system must not use excessive amounts of data. This is achieved using the technique of textising. Since, we convert all data to text data before crawling.

- Multiple parts of the system should work together seamlessly, creating an effect that the system is working as one. This is achieved by using the technique of message passing between languages.

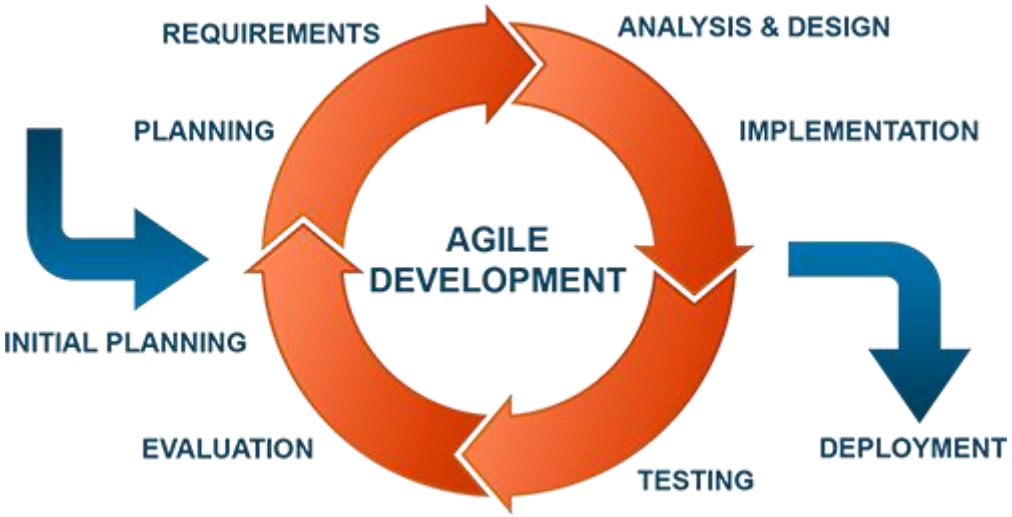
3.3 Overall Description of the Project

The aim is to create a power application that can crawl data in real time and analyse the results to produce intelligent data. The first part of the project will input the pre provided details of the person. The trusted will be defined in this step too. The information will be crawled from the trusted sources and added to the database. The information will be cleaned and removed of all the noise. The cleaned information will be processed to produce meaningful and human understandable format. This part will use natural language processing. The final report will be displayed in accordance with a pre defined tab based browser.

3.4 Design Documents

Process Models:

A process model is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required. Our project is implemented using Agile Development Model.



Agile Development Model:

- Welcome changing requirements, even in late development
- Working software is delivered frequently (weeks rather than months)
- Close, daily cooperation between business people and developers
- Projects are built around motivated individuals, who should be trusted
- Face-to-face conversation is the best form of communication (co-location)
- Working software is the principal measure of progress
- Sustainable development, able to maintain a constant pace
- Continuous attention to technical excellence and good design
- Simplicity—the art of maximizing the amount of work not done—is essential
- Self-organizing teams
- Regular adaptation to changing circumstance

Advantages:

1) Generates working software quickly and early during the software life cycle.

2) This model is more comfortable and requirements changing can be done less costly.

3) Testing and debugging is easy in initial iteration.

4) Feedback can be easily obtained from the development work that has been done.

5) Rapid delivery and deployment of useful software is possible.

Unified Modeling Language:

UML is a system for describing system at high level of abstraction. It is an industry-standard graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems. The UML (Unified Modeling Language) uses mostly graphical notations to express the OO analysis and design of software systems. UML moves from fragmentation to standardization and not dependent on any language or technology. UML uses graphical notation to communicate more clearly then natural language (imprecise) and code(too detailed). UML includes nine diagrams – each capturing a different dimension of a software system architecture which is:

a) Use Case Diagram

b) Class Diagram

c) Object Diagram

d) Sequence Diagram

e) Collaboration Diagram

f) State Chart Diagram

g) Activity Diagram

h) Component Diagram

i) Deployment Diagram

3.4.1 Use Case Diagram:

1) Use cases serve as a technique for capturing the functional requirements of a system.

2) Describes the typical interactions between the users of a system and the system itself, providing a narrative of how a system is used.

3) A use case consists of a set of one or more scenarios tied together by a common user goal.

4) A scenario is a sequence of steps describing an interaction between a user and a system; Some scenarios describe successful interaction; others describe failure or errors.

5) Users are referred to as actors; an actor is a role that carries out a use case.

6) An actor need not always be a person; it can also be an external system that is either automated or manual.

7) A use case diagram is like a graphical table of contents of the use cases for a system.

8) Use cases represent an external view of the system; consequently, they have no correlation to the classes in the system.

9) Actor: A role that a user plays with respect to the system, including human users and other systems. An external system needs some information from the current system.

10) Use Case: A set of scenarios that describing an interaction between a user and a system, including alternatives.

11) Include: a dotted line labeled <<include>> beginning at base use case and ending with an arrows pointing to the include use case. Include relationship occurs when a chunk of behavior is similar across more than one use case. Use "include" instead of copying the description of that behavior.

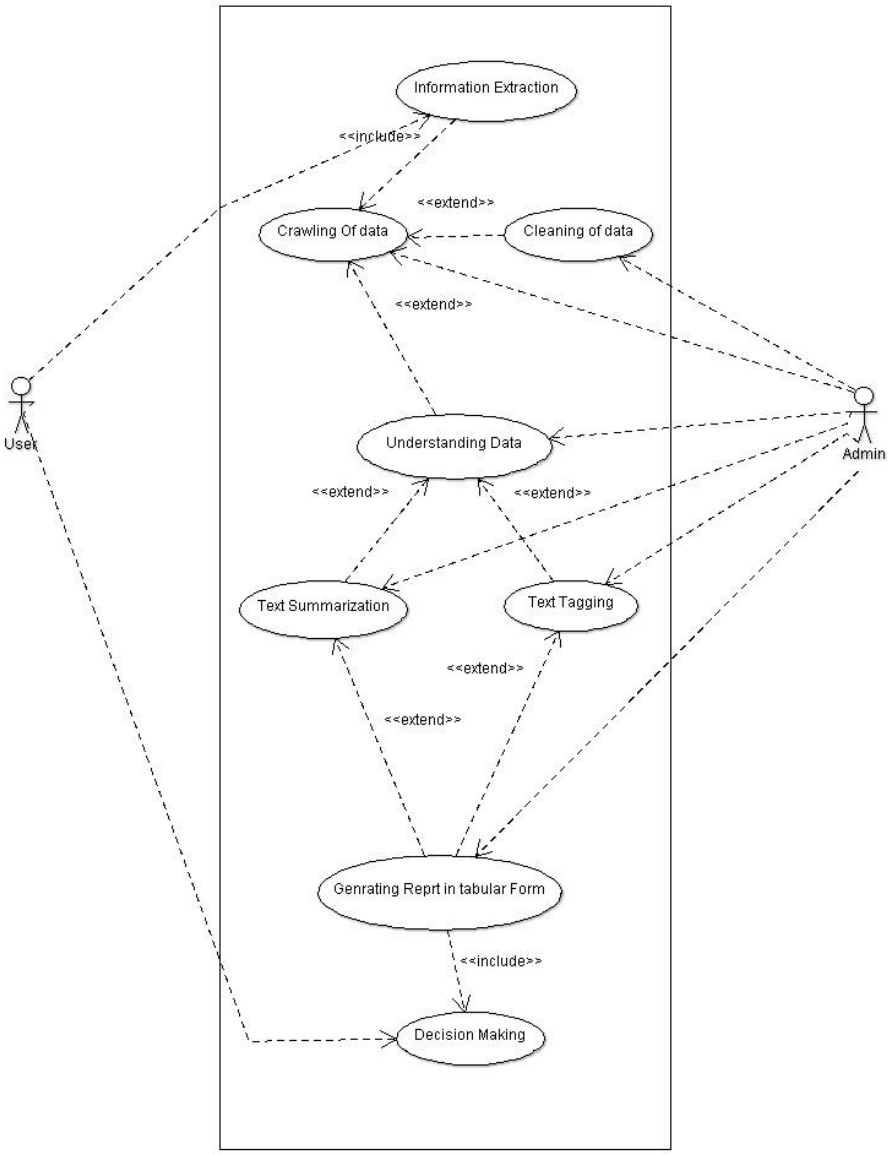
12) Extend: a dotted line labeled <<extend>> with an arrow toward the base case. The extending use case may add behavior to the base use case.

13) System Boundary: rectangle diagram representing the boundary between the actors and the system.

14) Association: communication between an actor and a use case; Represented by a solid line.

15) A use case diagram shows the use cases, the actors, and the relationships between them so that they can serve as a starting point for writing software validation test cases.

Use Case Diagram



3.4.2 State Diagram:

1) Describe all the possible states an object can assume and how the object's state changes as a result of events that affect the object.

2) State Diagrams are drawn for a single class to show the lifetime behavior of a single object.

3) State Diagrams are good for describing the behavior of an object across several use cases.

4) A state diagram consists of states and transitions.

5) Note that a state diagram is NOT a set of processes connected by lines representing data input and output.

6) A state is characterized by the current values of an object's attributes and its name reflects some ongoing activity or state.

7) A transition indicates a movement from one state to another because an event has occurred; this transition changes one or more attribute values of the class.

8) Action – processes associated with transitions that occur quickly and are not interruptible.

9) Activity – processes associated with states that may take a while and may be interrupted by events.

10) Event – A stimulus that causes a transition or a self-transition to take place from one state to another.

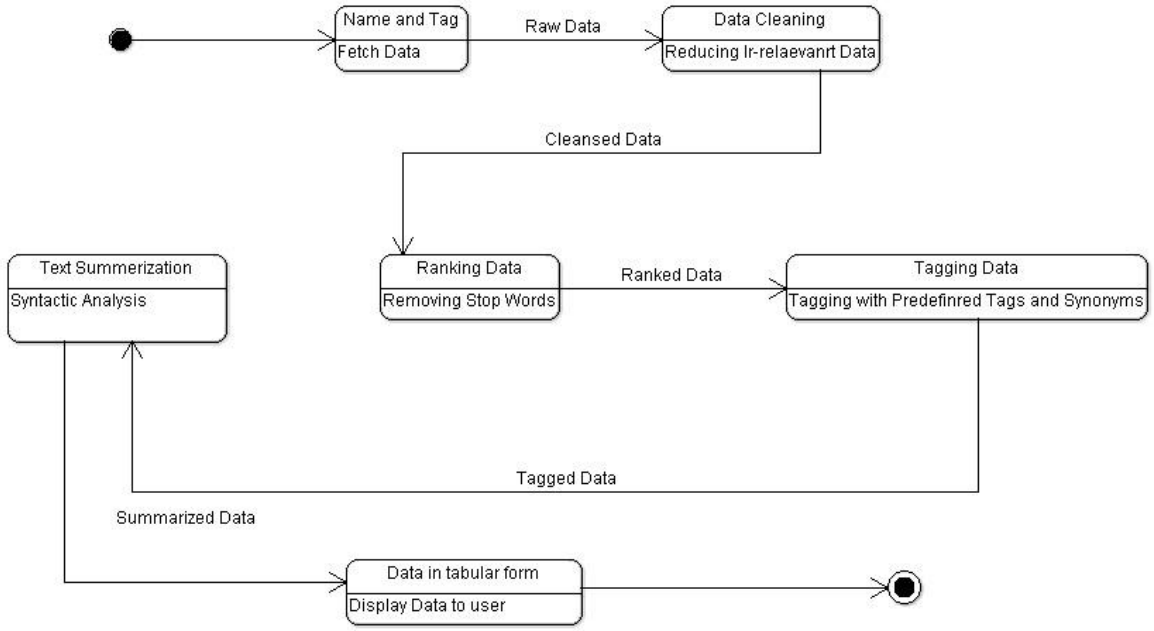
11) Guard - A logical condition that returns "true" or "false".

12) Super State - A state that is itself a collection of states.

13) A filled circle followed by an arrow represents the object's initial state.

14) An arrow pointing to a filled circle nested inside another circle represent the object's final state.

State Chart Diagram



3.4.3 Activity Diagram:

1) Activity diagrams helps to describe the flow of control of the target system, such as the exploring complex business rules and operations, describing the use case also the business process. It is object-oriented equivalent of flow charts and data-flow diagrams (DFDs).

2) Used to represent the behavior of a system in terms of activities and their precedence constraints.

3) Compared to flowchart diagrams because …

- They can be used to represent control flow (order in which operations occur)
- They can be used to represent data flow (objects exchanged among operations)

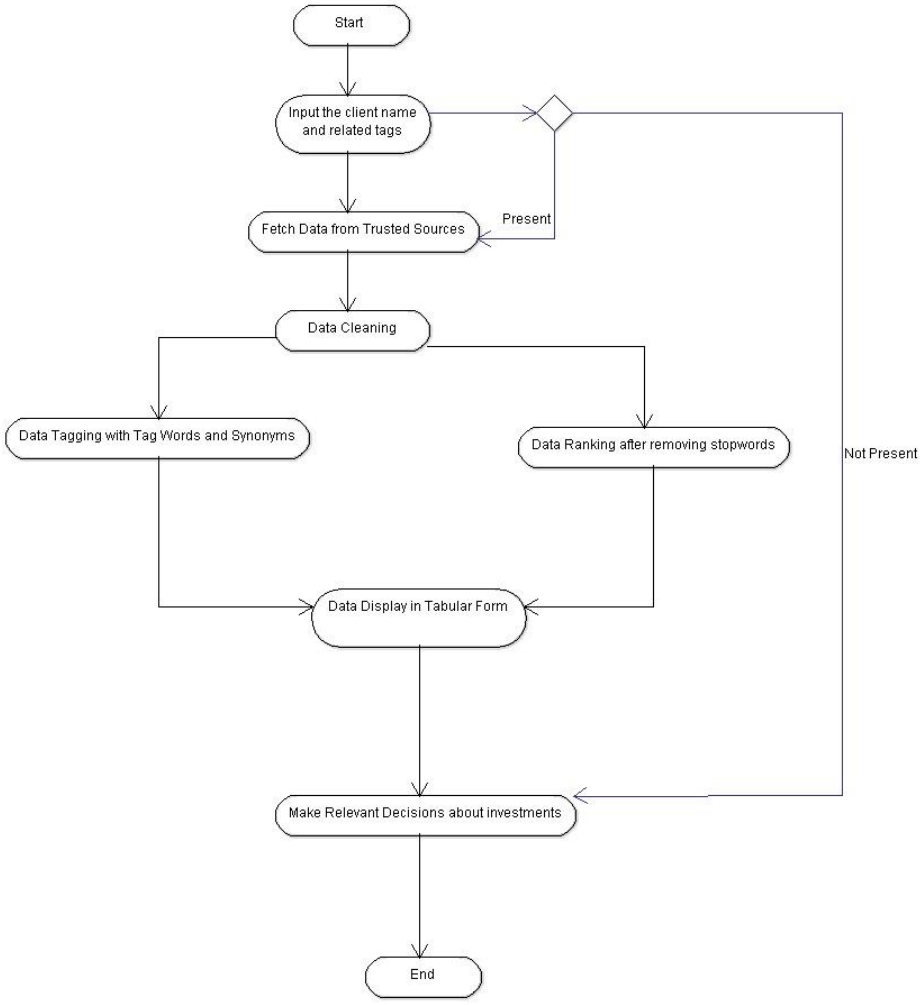
4) The completion of an activity triggers an outgoing transition which may initiate another activity.

5) Serves as a technique to describe procedural logic, business process logic, and work flow is similar to a flowchart except that it can also show parallel behavior.

6) States the essential sequencing rules to follow, thereby allowing concurrent algorithms to be used.

7) Consequently, an activity diagram allows whoever is doing the process to choose the order in which to do certain things.

Activity Diagram



3.4.4 Data Structures and Algorithms:

Graphs: In the most common sense of the term, a graph is an ordered pair G = (V, E). It comprises a set V of vertices or nodes or points together with a set E of edges or arcs or lines, which are 2-element subsets of V (i.e., an edge is related with two vertices, and the relation is represented as an unordered pair of the vertices with respect to the particular edge). To avoid ambiguity, this type of graph may be described precisely as undirected and simple.

V is a set together with a relation of incidence that associates with each edge two vertices. In another generalized notion, E is a multi set of unordered pairs of (not necessarily distinct) vertices. Many authors call this type of object a multi graph or pseudo graph. We have made clusters in graphs to tag sentences.

Multi Lists:

In a general multi-linked list each node can have any number of pointers to other nodes, and there may or may not be inverses for each pointer. This helps save unordered and unnumbered data.

Data Cleansing Algorithms: Data cleansing, data cleaning or data scrubbing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. Used mainly in databases, the term refers to identifying incomplete, incorrect, inaccurate, irrelevant, etc. parts of the data and then replacing, modifying, or deleting this dirty data or coarse data.[1] Data cleansing may be performed interactively with data wrangling tools, or as batch processing through scripting.

After cleansing, a data set will be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores.

Syntactic Analysis: Syntactic Analysis is the process of analysing a string of symbols, either in natural language or in computer languages, conforming to the rules of a formal grammar. In some machine translation and natural language processing systems, written texts in human languages are parsed by computer programs. Human sentences are not easily parsed by programs, as there is substantial ambiguity in the structure of human language, whose usage is to convey meaning (or semantics) amongst a potentially unlimited range of possibilities but only some of which are germane to the particular case. So an utterance "Man bites dog"

versus "Dog bites man" is definite on one detail but in another language might appear as "Man dog bites" with a reliance on the larger context to distinguish between those two possibilities, if indeed that difference was of concern. It is difficult to prepare formal rules to describe informal behaviour even though it is clear that some rules are being followed.

3.5 Risk Analysis and Mitigation Plan:

Risk analysis is a technique used to identify and assess factors that may jeopardize the success of a project or achieving a goal.

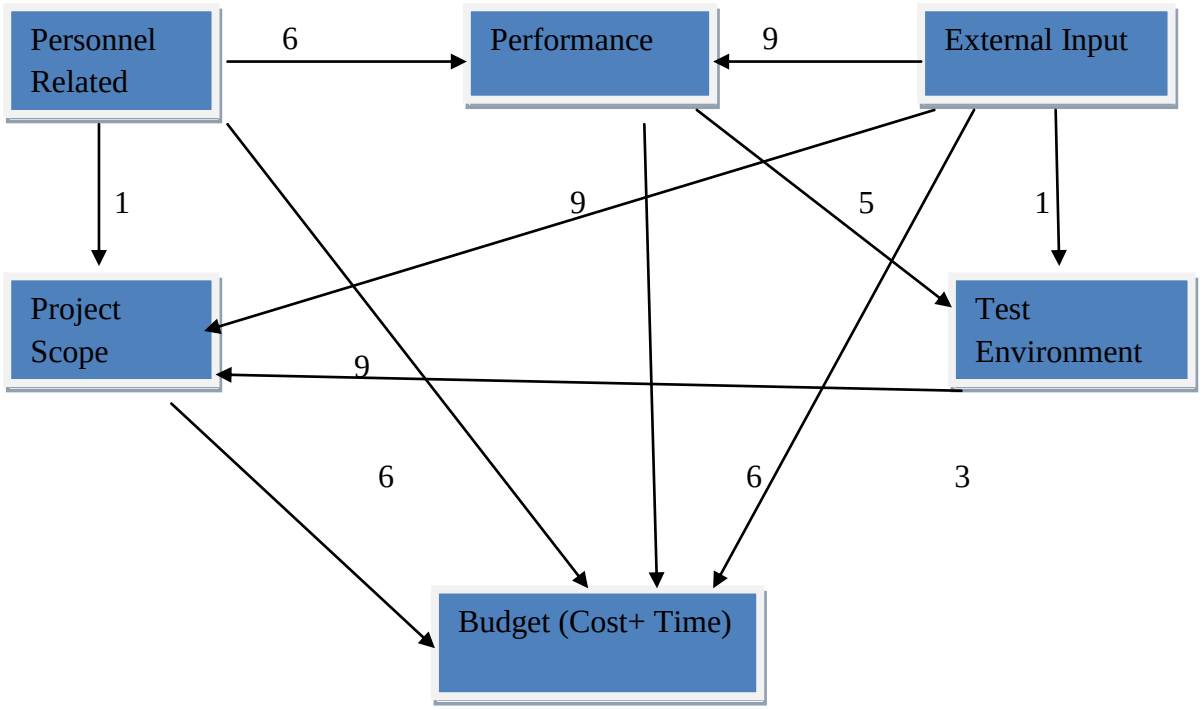
Risk analysis can be defined in many different ways, and much of the definition depends on how risk analysis relates to other concepts. Risk analysis can be "broadly defined to include risk assessment, risk characterization, risk communication, risk management, and policy relating to risk, in the context of risks of concern to individuals, to public- and private-sector organizations, and to society at a local, regional, national, or global level.

Actuarial science has used probabilities to measure risk for more than a hundred years, probability risk assessment as a specific mode of inquiry was initially developed to analyze engineering risks such as nuclear power plants and the space shuttle. More recently, it has also been applied to other areas, such as business, climate change, health risks, food safety and security. Especially with the increasing i1mportance of terrorism, game theory has become a quantitative tool to analyze the risks of intelligent adversaries who seek to do harm against a system or people.

Risk Management Description

Risk ID	Description of Risk	Cause	Risk Area	Risk Probability	Risk Impact	RE
1	Time for Crawling Information from each website	The amount of data present on the internet	Processing based on user defined Input	Probability: Medium(3)	Impact: High(5)	RE: 15
2	The correctness of Natural Language Processing	The complexity of sentences and the presence of useless data	Processing on the crawled data	Probability: Medium(3)	Impact: High(5)	RE: 15
3	Summarising the crawled and processed information	The presence of homonyms and complex punctuations.	Displaying the processed data	Probability: Low(1)	Impact: Low(1)	RE: 1
4	No valuable data is found regarding the input.	Unforeseen Risks	User Input	Probability: Low(1)	Impact: High(5)	RE: 5

Weighted Inter-Relation Graph



Risk Area Wise Total Factor

S.N.	Risk Area	# of Risk Statements	Weights (In +Out)	Total Weight	Priority
1	Budget	5	9+9+6+3+6	16	2
2	Performance	4	6+9+6+5	27	1
3	Project Scope	4	9+9+6+1	25	4
4	Eternal Input	4	9+1+9+3	22	5
5	Personnel Related	3	1+6+9	16	6
6	Test Environment	3	9+5+1	15	3

4. Implementation and Testing

4.1 Testing

4.1.1 Testing Plan

Test Plan

Requirement Testing	Yes	Output generated gives a concise report	Final output of program
Unit	Yes	Independent extraction from news websites, Wikipedia and LinkedIn	Independent working of each component
Integration	Yes	Final output of report correctly merged	Final output of program
Performance	Yes	Data extracted is accurate but may leave some useful information	Extraction part of the program
Stress	Yes	Time taken in data extraction depends on internet speed	Crawling part of the program
Security	No	Data kept in text files in readable format	NA
Load	Yes	Reports of different person generated each time	Crawling part of the program
Volume	No	Text data does not require large size	NA

Test Team Details

Test Engineer	Anshit Agarwal	Collect data on which program is to be tested
Test Engineer	Karan Dhingra	Check the accuracy of report generated

Test Schedule

Collection of input data	12/20/2015	12/20/2015	1 hour	Names of people searched in news articles and famous personalities known
Check validation of output generated	12/20/2015	12/20/2015	1 hour	Verified whether the output generated gives enough information

4.1.2 Component decomposition and type of testing required

Component Decomposition and Test Identification

1.	Data crawling	Unit, Stress, Load	White Box
2.	Data extraction from crawled data	Unit, Performance	Black Box
3.	Reading data from data extracted	Requirement, Unit	Black Box

Test Case for Components

1.	Data crawling	Person Name	Text file	Pass
2.	Data extraction from crawled data	Person Name	Text file with reduced data	Pass
3.	Reading data from data extracted	Person Name	Output displayed on website	Pass

1) Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied to virtually every level of software testing: unit, integration, system and acceptance.
2) White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing software that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing).
3) Although software testing is itself an expensive activity, yet launching of software without testing may lead to cost potentially much higher than that of testing, especially in systems where human safety is involved.
4) We should test the program's response to every possible input. It means we should test for all valid and invalid inputs. We have also not considered invalid inputs where so many combinations are possible. Hence, complete testing is just not possible, although, we may wish to do so.

4.1.3 Limitations of the solution

The final report generated may not contain some of the useful information and th report may also have some information which is not related to the person because of extra data available on websites.

5. Findings and Conclusions

5.1 Findings

Risk analysis is a difficult problem. Since it is concerned more with finance than computing, the problem has not been extensively explored by corporations. The risk of investing in any individual requires mitigation. To mitigate this risk, companies employ many employees whose work is to read the pre-present data about the individuals and decide whether investing in them is profitable or not. We are trying to automate the task.

While completing this task, we faced several problems and coined solutions for those problems. Here is a list of a few.

1. The crawled data might not have been relevant. To make this information relevant, we cleaned the data and tagged the sentences using predefined tags.
2. The importance ranking of the sentence is a debatable topic. To solve this problem, we ranked the sentences after removing stop words. A list of stop words was crawled from reliable sources.
3. Crawling websites like LinkedIn and Google is not allowed. Since they contain lot of valuable data, we used a method of textising the complete data.

5.2 Future Work

1. The crawling accuracy of the data can be increased, since a lot of irrelevant data is crawled right now. This also leads to increased data usage.

2. More trusted websites like Reuters and PBS can be added to the data bank.

3. The attributes tags can be made machine learnable. This can help the system get better with time and reduce human intervention in making decisions.

References:

1. Yue Chen, (2010). "Natural Language Processing in Web data mining"

2. Mills M.T., Bourbakis N.G., (2013). "Graph-Based Methods for Natural Language Processing and Understanding—A Survey and Analysis"

3. Chris Biemann, (2006). "Chinese Whispers - an Efficient Graph Clustering Algorithm and its Application to Natural Language Processing Problems"

4. Jonas Sjobergh, Kenji Araki, (2008). "Extraction based summarization using a shortest path algorithm"

5. Martin Hassel, Jonas Sjobergh, (2009) "Towards Holistic Summarization ?Selecting Summaries, Not Sentences"