

Final Report
Level - 4
Intelligent Tour Guide System - ITGS

By
Patriots

124111R	Mohamed.M.F
124016F	Asir.A.B.M
124119A	Nifras.M.T.M

Faculty of Information Technology

University of Moratuwa

July 2017

Final Report
Level - 4
Intelligent Tour Guide System - ITGS

By
Patriots

124111R	Mohamed.M.F
124016F	Asir.A.B.M
124119A	Nifras.M.T.M

Supervised By: Dr. (Mrs.) A Thushari Priyangika silva
Senior Lecturer,
Department of Computational Mathematics
Faculty of Information Technology,
University of Moratuwa

Faculty of Information Technology
University of Moratuwa

July 2017

Abstract

Information overloading has posed significant challenges on identifying required information for travelling, based web site articles, user reviews, comments, tweets and posts. When the tourist are on tour, they need to make a decision on a certain place. Whereas they have to go through the all kind of tourism based websites to read more such reviews of previous visitors. There are lack of automate reviews aggregating systems in tourism industry of Sri Lanka. Current approaches of tour guides such as Trip Advisor, Wonder of Asia, and Srilanka.travel provide only an isolated view as they could have only a single source, fail to address semantic heterogeneous data sources.

In order to overcome these difficulties an automated system which integrate multiple heterogeneous data such as reviews, tweets, comments, posts developed Thus system could summarize reviews and offer real time information for effective decision making. Therefore to make perspective of the reviews from on-line, crucial first step is to understand the semantics of those reviews. In our system textual information provided by reviews from web sources. Those reviews were extracted by using web crawlers. Extracted reviews were preprocessed using text mining techniques. And for structuring of information, an ontology based approach is used by developing Travel.owl for this system and Sparql queries for semantic web search. With developing subject classifier and sentiment classifier to provide for subject classification and aggregate rating in the Intelligent Tour Guide System to get the overall rating for a tourist place. By using ITGS, users can gain fascinate and comport of their tour. And with the function of aggregating rating, users able to sort the attraction places when their on tour. This system will support to popularize the tourist places in Sri Lanka. Therefore ITGS can support to increase the economy enhancement of the nation.

Table of Contents

Abstract.....	i
Tables of figures.....	v
1. 0 Introduction.....	1
1.1 Introduction.....	1
1.2 Aim and Objectives.....	4
1.1.1 Aim.....	4
1.1.2 Objectives.....	4
2 Others work.....	5
2.1 Introduction.....	5
2.2 Free text retrieval.....	5
2.2.1 Structure of data sources.....	5
2.2.2 Challenges in free text retrieval.....	6
2.2.3 Extracting techniques.....	6
2.2.4 Fetching techniques.....	8
2.3 Natural Language Processing.....	9
2.3.1 Infrastructure for language processing.....	9
2.4 Text classification.....	13
2.4.1 Text classification process.....	13
2.4.2 Methods and techniques for text classification.....	14
2.4.3 Challenges in text classification.....	16
2.5 Mapping to Ontology.....	18
2.5.1 Ontologies for information extraction.....	18
2.5.2 Developing Ontology.....	20
2.5.2.1 Ontology languages.....	20
2.5.2.2 Ontology editors/tools.....	20
2.5.2.3 Aligning relations.....	21
2.5.2.4 Ontology population process.....	21
2.6 Aggregating ratings.....	22
2.7 Summary.....	22

3	Technology Adapted.....	23
3.1	Introduction.....	23
3.2	Used technologies and concepts.....	23
3.2.1	Used technologies for text retrieval.....	23
3.2.2	Used technologies for NLP.....	23
3.2.3	Used technologies for mapping ontology.....	25
3.2.4	Used technologies for text classification.....	26
3.3	Summary.....	26
4	Approach for ITGS.....	27
4.1	Introduction.....	27
4.2	Text retrieval approach.....	27
4.3	Language engineering approach.....	28
4.4	Semantic web approach.....	29
4.5	Text classification approach.....	29
4.5.1	Subject classifier.....	29
4.5.2	Sentiment classifier.....	30
4.6	Inputs and Outputs.....	30
4.7	Used tools and technologies.....	31
4.8	Summary.....	31
5	Design and Implementation.....	32
5.1	Introduction.....	32
5.2	Top level architecture system.....	32
5.2.1	Annotated GATE pipeline.....	33
5.2.2	Design of text classification.....	38
5.2.3	Developing ontology.....	40
5.2.3.1	Mapping keyword to Ontology Class.....	40
5.2.4	Aggregation rating.....	40
5.3	Summary.....	42
6	Experiment.....	43
6.1	Subject Classification.....	43
6.1.1	Precision and recall.....	43
6.1.2	Training data set.....	44
6.2	Predicting Sentiment value.....	44

7	Results and Discussion.....	46
7.1	Results.....	46
7.1.1	Subject Classification.....	46
7.1.2	Predicting Sentiment value.....	47
7.2	Discussion.....	48
8	Conclusion	49
	References.....	51
	Appendix A- Individual contribution to the project.....	54
	Appendix B- Some details of implementation.....	57
	Appendix C- Declaration.....	66

Table of Figures

Figure 2.1: UIMA transform unstructured information to structured information.....	10
Figure 2.2: Parse tree relations of NLTK process.....	11
Figure 2.3: Co-occurrence network diagram of KH Coder.....	12
Figure 2.4: SVM classification method.....	15
Figure 2.5: KNN classification method.....	16
Figure 5.1: Top level architecture of ITGS.....	32
Figure 5.2: Annotated GATE pipeline.....	33
Figure 5.3: Document with tokenizer in GATE GUI.....	34
Figure 5.4: Document with sentences in GATE GUI.....	35
Figure 5.5: Example format of Jape rule.....	36
Figure 5.6: GATE pipeline.....	37
Figure 5.7: Annotations using GATE and Classifiers.....	38
Figure 5.8: Ontology class hierarchy.....	39
Figure 5.9: Process flow of implementation of semantic web in ITGS.....	40
Figure 5.10: Aggregation rating for reviews.....	41
Figure 6.1: Actual value and Predicted value in linear regression.....	44
Figure 7.1: Precision vs Training Instances.....	46

CHAPTER 1

1.0 Introduction

1.1 Introduction

Today tourism is the fastest growing industry in the world, “With the year 2014 recording over one billion tourist, generating US\$ 1.2 trillion revenue and accounting for almost 10 percent of the global GDP while 1 in 11 jobs worldwide were tourism related”. And also tourism is the big role in the Sri Lankan Economy. Employment generations obtain much rewards from Sri Lankan tourism industry. Sri Lanka attracts due to the geographical spread of nature and places. “Sri Lanka is truly blessed with one of the most diverse and unique natural environments in an island of our size” [1]. And Sri Lanka has unique tourism destination places. Thus most number of tourists willing to visit Sri Lankan attractive places. “During year 2015 up to November, Sri Lanka Tourism recorded 18.1% growth on arrivals with 1.5 million arrivals. The total tourism revenue generated was US\$ 2.2 billion” [1]. Due to increase in arrivals of tourists, they need proper tour guide system to make their successful tour.

Mainly Tourism information are obtained from newspapers and websites in Sri Lanka. To enhance the Sri Lankan tourism from traditional based tourism guide into a modern technologies based tourism guide, and to transform Sri Lanka from “a tourism country” into “top tourism country”, it is most important to emerging Internet and Artificial Intelligent techniques to upgrade the tourism services related systems. With the rapid development of tourism, the requirements of the community increase without stop on the comprehensive ability of tourist guiding workers.

According to the tourism domain, World Wide Web contains a large number of reviews. They provide details regarding tourism places and attractions. Those reviews express ideas about the places, experience, activities, events, fees and lot of other useful information regarding an attraction. And due to the availability of large number of reviews, it is not suitable to go through them manually.

Tourist guides are playing major role in tourism industry. Despite foreign tourist are afraid to visit due to trust worthiness of the tour guide workers. Thus tourists are prefer proper tour guide system behalf of tour guide workers. There are some tour guide system available. But they are not intelligent to guide in an efficient way. Because of those systems are not consider the variety of review data sources. Traditional virtual guide training in the beginning mainly relied on books to teach, using words, pictures, scenery, model, sand table and other ways to display Places and environment. The teaching method was so limited that the result is not ideal. Travel and traffic data, due to the rapid development of information and communication technologies are becoming increasingly and more fast accessible to the users.

To make intelligent tour guide system, should have to include Big Data analysis of the human representation of the tourism arena. Regarding to Big Data analysis, the system should go through each and every different types of review data sources from World Wide Web. The data can be obtain from following ways,

- Tourism based website articles
- User reviews in websites
- Posts and comments
- Tweets
- Wikis
- Blogs
- Google Map

Overloading these kind of textual information led to inconvenient, because of complexity in tourism domain. When a visitor is on tour, he/she makes a decision on certain place, whereas that particular visitor has to go through several on-line sources. Collecting the all kind of textual information is not an easy task. There are lack of automate review aggregating systems in tourism industry. And current tourist based systems are not analyze all kind of review data sources and integrate into one single system. Therefore to make perspective of the reviews from on-line, crucial first step is to understand the semantics of those reviews.

“With additional numbers expected to be promoted most popular tourism sites visited by both local and foreign tourists are going get overcrowded. If such sites are not improved on a proper plan and good management, the value to the visitors will not remain for long” [1]. Therefore the reviews and experience of previous visitors are more helpful to the tourists for finding the best places.

To overcome the above mentioned problem, we proposed a web based Intelligent Tour Guide System (ITGS). ITGS will provide efficient tour guide to users. Travel domain is spread among multiple aspects, but ITGS mainly focused on attraction places of Sri Lanka. The main areas of research such as Semantic Web and Ontology Modelling, Data Mining, Natural Language Processing (NLP) and Machine Learning. The aim of the research is to propose a solution for a major issue in tourism industry regarding extracting information from World Wide Web and find the attraction places efficiently. ITGS divided into following main categories.

In first free text and reviews related to the attraction places of Sri Lanka were retrieved from different web sources. Then retrieved textual information were preprocessed, annotated, and store in data storage. After that developing ontology and mapping relationship between attraction places and their features. Extracted information should be properly structured to facilitate querying. Therefore decided to use an ontology based approach for structuring of information. After that created module for text classification which under this section there are two main parts, Subject classifiers and Sentiment classifiers. The main idea of aggregating rating is calculate overall sentiment value for each attraction places.

1.2 Aim and Objectives

1.2.1 Aim

The main aim of this project is to develop web based Intelligent Tour Guide System which can provide effective and efficient tour guide to the visitors, thus visitors can find attraction places via integrated multiple information sources.

1.2.2 Objectives

The objectives of the research project are,

- Extracted information from multiple sources
- Noise removal filtering extracted information
- Preprocessing and classification of data
- Develop an Ontology and Automated Mapping key words to ontology classes
- Develop feature engineering for subject and sentiment classifiers
- To analyze reviews sentiment
- Provide suggestions according to aggregating rating

In this report, Chapter 2 describes the others work that how others achieve the solutions from different directions to the above mentioned problem and comparison and identify some pitfalls of others approach. Technology Adapted is described in Chapter 3. In Chapter 4 describe the approach for ITGS, the section describe that how to adapted technologies used for each scope our project. Chapter 5 describes Design and Implementation of ITGS, which mentioned the main top level architecture system and described the processes of relevant modules. Finally Discussion are put forward in Chapter 6.

CHAPTER 2

2.0 Others Work

2.1 Introduction

The existing tools and technologies that relevant to ITGS in this chapter. There are four main parts such as Free Text Retrieval, Natural Language Processing, Ontology Developing and Aggregation rating which are referred from literature reviews.

2.2 Free text retrieval

In information extraction, the first operation is free text retrieval, and it is very important operation for accuracy of the information which extracted from the web articles. Reviews and articles are representing different formats in the web, and extracting information is depending on those formats. Nowadays most of the web sources are in HTML format, and encrypted. In this section, discussed about free text retrieval based on others works.

2.2.1 Structure of data sources

Commonly information extraction (IE) form web sources are performed by using wrappers. Main purpose of wrapper is access the HTML document and converts into suitable format, usually XML format. Web wrappers made based on the different kind of Web pages. Giacomo Fiumara covered the types of the web pages and information extraction tools in his paper “Automated Information Extraction from Web Sources”. There are 3 type web pages are available in the web [7], Unstructured Pages, Structured pages and Semi-structured pages.

- **Unstructured Pages**

These pages are written using natural language and there is no structure in these kinds of pages. These pages also known as free-text documents.” only in-formation extraction (IE) techniques can be applied with a certain degree of confidence” [7].

- **Structured pages**

These kinds of pages are derived from structured data sources. Using some simple techniques, the extraction process can be performed. Those techniques are based on syntactic matching [7].

- **Semi-structured pages**

Semi-structured pages do not conform to formal structure, and these kinds of pages are intermediate position between structure page and unstructured pages.” Commonly information extraction techniques are based on the presence of special patterns” [7].

2.2.2 Challenges in Free text retrieval

Emilio Ferraraa, Pasquale De Meob, Giacomo Fiumarac, and Robert Baumgartnerd, they discussed about challenges in free text retrieval in their survey “Web Data Extraction, Applications and Techniques” [2]. Some of those challenges are discussed below.

Web data extraction methods/techniques need human involvement to provide best performance. If human involvement is increase in the techniques, those techniques are not good. Thus have to reduce the human involvement as much as possible. Reducing human efforts is big challenge in web data extraction techniques. Web Data Extraction techniques should be effective and efficiency, otherwise techniques will useless. To increasing the effective and efficiency techniques should be mange to extract the large amount of data within the short time of period. Next big challenge is, technique/methods should not violate the privacy of the users, when extracting the data from social related web or social profile related data. Nowadays structure of the web related pages are changing day by day, this led to big problem when the data extract in timely manner [5].

2.2.3 Extracting Techniques

Web data extraction techniques extract the data from web sources and extracted date should store effectively and efficiently with less human involvements. Some of the extracting techniques are depend on application which need to extract the data and others are independent to applications. There are many extracting techniques available, and, discussed some of those techniques bellow.

- **Web Wrappers**

Main purpose of wrapper is extracting structure data form unstructured or semi structured web related pages. Based on the Wrapper's work, wrapper can divided into 3 stages, also known as life- cycle for web wrappers.

- Wrapper generation: According to some technique/s, a particular wrapper should defined.
- Wrapper Execution: Wrapper continuously runs and extracts the information.
- Wrapper maintenance: wrapper should change as structure of data source change.

- **Tree-based techniques**

Semi-structured nature of web ages can be represented as labeled ordered rooted trees, in this techniques labeled represented as Markup language tags and nesting of elements in a web page represent as tree hierarchy. Assign the web page into labeled ordered rooted tree is known as DOM (Document Object Model). XPath data extracting technique is adaption of DOM [5].

- **Hybrid systems: learning-based wrapper generation**

This technique works while using wrapper generation platforms and learning-based wrapper induction system. Template-based matching is one of the best examples for Hybrid system. This system works based on wrapper generator and wrapper induction, also system use the prior knowledge which learned previous page of extracted data. Spatial reasoning technique is another example for Hybrid Systems. This technique works based on X-Y cut OCR algorithm [5]. Thus it can able get the rendered version of a web page using this technique.

- **XML based data extractions**

Main purpose of this kind of approaches is extract the semi structured data form particular web page and transform into structured format, this process known as rich representation. There are many XML based data extraction techniques steps, Web Site Navigation is first step in XML based data extractions process, Mainly two types of web pages can identify in the HTML web pages[13]. These types of HTML pages that we want to extract the data and contain hyperlinks for redirect the other navigational pages or target pages .By using automated crawler for retrieve the target page from particular website, and crawler is guided by a rule based configuration file, which tells the crawler where to start and which hyperlink have to follow. After the web site navigation process, the followed 7 steps, were Data Extraction, Hyper-link Synthesis, Structure Synthesis, Data Mapping, Data Integration, Data Validation, and Data Export.

2.2.4 Fetching Techniques

In order to fetch a particular website page, there must have an HTTP client, and that HTTP client will send HTTP request for that particular web page, after sending the HTTP request, it will read the given response .And should have an timeout for ensure that particular client not spent much of time on server or reading large sizes pages, so that we have to restrict the page size around 10-20KB.client should parse the headers of the response for status codes and redirections. During the fetching techniques Exception handling and Error checking are very important process due to we have to deal with lot of remote servers using the similar code. Some modern languages, such as Perl and Java, provides interface for fetching web pages [25].

2.3 Natural Language Processing

Text is everywhere, it provides our various social feeds, fills up our inboxes and commands attention, for all human text representations to get the actual and accurate idea about that we doing Natural Language Processes. NLP focuses wide range of methodologies to decipher and reduce the ambiguities in human representation of languages, it may considered following disciplines of text enhancing processes, automatic summarization, entity extraction and relation extraction and also provide disambiguation of human representation as well as natural language understanding and text recognition.

2.3.1 Infrastructure for language processing

In Natural Language Processing, it may can sub divide parts regarding to its basic functionality such as storing information, displaying information, loading processing resources, enhancing processing resources and passing information by data level among machine and other processes. And many more language enhancement processes can be found in language infrastructures. It helpful to researchers and developers, by conducting parameterization data and set the parameters which is most suitable for the particular task and get the life easier using the infrastructure unless highest optimization workloads that contain some low level programming knowledge also required [11]. In this paper mention some of language infrastructures which used to enhance text enhancement for human language representations, and recognize the patterns of the text and the meaning of it, whereas we mention below their advantages and disadvantages in following sub sections.

- **Untrusted Information Management Architecture (UIMA)**

Untrusted Information Management Architecture is a software system which is focuses on analyze the large volume of untrusted text information, audio and video information enhancement and discover knowledge and implementation for C++ and java. It can be contain plain text and recognize certain domain level entities, such as organizations, names, places, relations, likewise. Different level of users get benefit using UIMA in different flavors, if it complex task for analysis then it can be different modules sections used to trigger the actual comfort from the UIMA. It support different level of plugins and it provides collaborative working environment.

UIMA application focuses analysis outcomes and does not worry about the actual details how the doers or annotators work together for picking outcomes. This framework supportive for orchestration multiple annotator's integrations. And the goal of UIMA is convert unstructured plain data to structured data by conducting orchestrating of multiple layer level analysis techniques to identify relations or certain entities for build the relation among unstructured and structured data or information [15].

The figure in below mention the UIMA intermediate the unstructured information with the structured information.

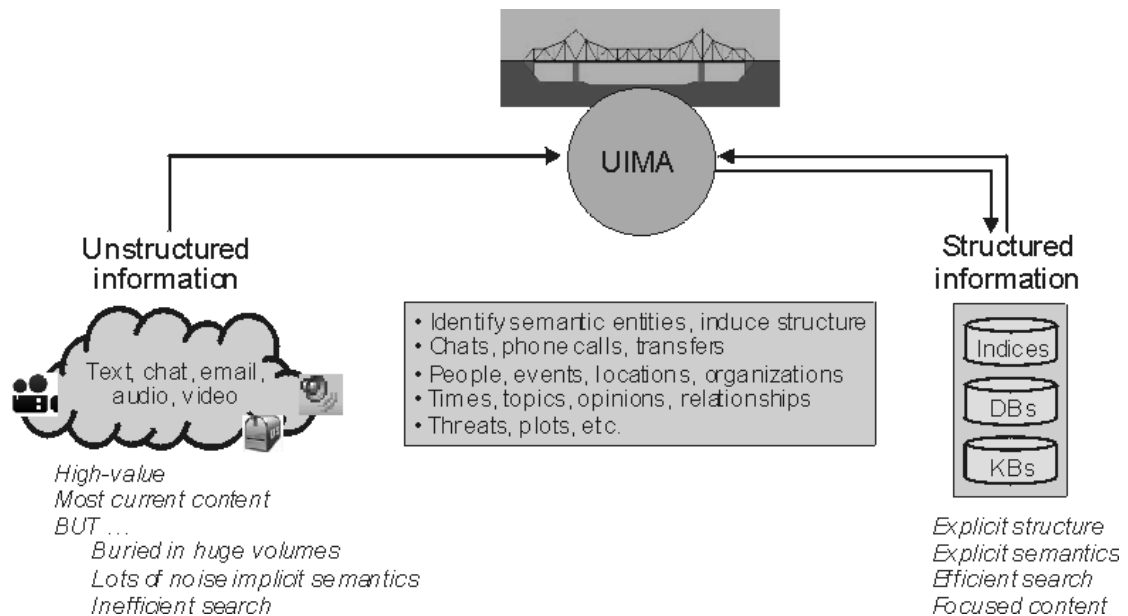


Figure 2.1: UIMA transform unstructured information to structured information [15].

UIMA is a framework mainly focused on untrusted data such as text audio and video. The case is it is not deeply focused the text mining real scenarios and not good for mapping recognize entities in various circumstances. And cannot sure for guarantee context of special rules.

- **Natural Language Toolkit (NLTK)**

Natural Language Toolkit, it is a framework for developing NLP tasks where python language can be used and enhance the human language representations. It supports for trivial workloads in NLP processes such as recognizing named entities, tokenizing process, display parse tree, stemming the main text/word Part Of Speech Tagging and also text classifications from sentiment polarity estimating [2].

Below figure shows the parse tree relations of NLTK process.

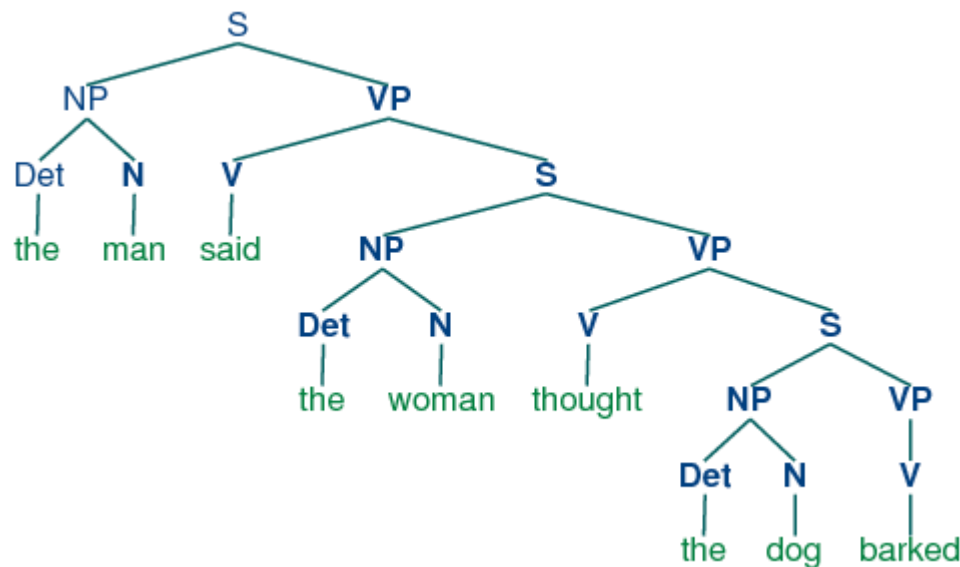


Figure 2.2: Parse tree relations of NLTK process [2].

And also we can find there some problems with NLTK which is depend on more data it needs more data for supervised training set data otherwise results may go wrong.

- **TAMS Analyzer**

This framework for Macintosh Operating Systems it can be identified themes of plain text in various repositories or areas of web pages. Tams Analyzer based on programs that collaborate with TAMS for enhance the text analysis such as extraction, analysis and save it relevant coded format [10].

- **LingPipe**

LingPipe is a framework for classification of the text by subjective and objective phrases. It could differentiate positive reviews and negative reviews separately. This framework used in classifier models for hierarchical techniques [14].

LingPipe deep with text classification only and here no deep works or methodologies for text recognition and provide rules and checking the results depend on the specific rule set.

- **KH Coder**

This is an application text analysis with quantities manner, and text mining and corpus linguistics. It can enhance the analysis by providing self-organizing map, scaling, network diagrams and also clustering analysis of the information and correspondence behavioral analysis. It could use back end techniques and tools such as POS tagger, stemmer (Snow ball) R and MySQL db [26].

Below shows the co-occurrence network diagram of KH Coder application.

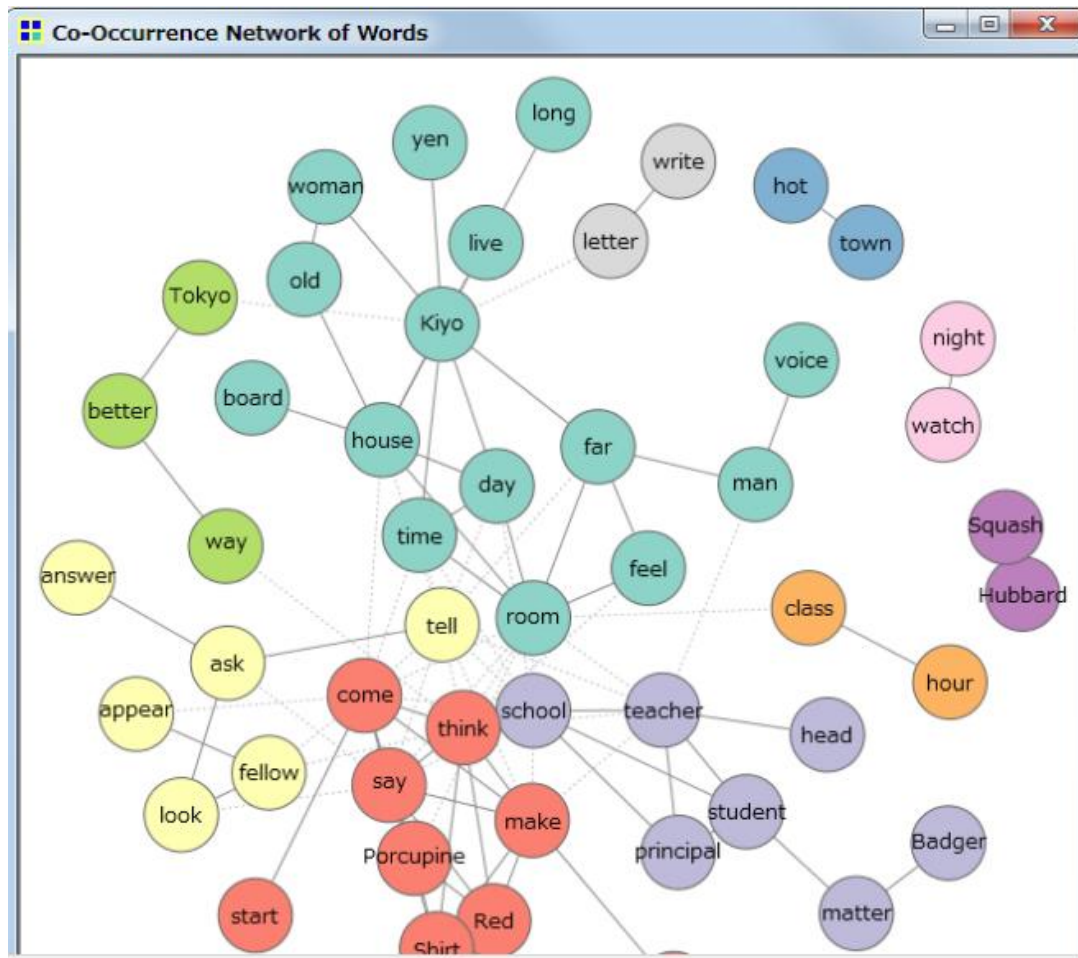


Figure 2.3: co-occurrence network diagram of KH Coder [26]

- **Sementria**

It is a tool based in Microsoft Excel for collaborative social details analysis. Most commonly used for tweets and Facebook post analysis. And its combination from manual and automation, we need to give data as an input manually. User needs to feed data to the system through an excel data sheet [20].

- **Gensim**

It is a text analysis tool platform for python programs. It can be used for statistical semantics and enhance the plain texts. It contains unsupervised algorithms for identifying the latent semantic text specific analysis by going through text word statistical of the co-occurrences methods with in a corpora or corpus of training data [18].

2.4 Text Classification

Text classification is the task for assigning any kind of topic category to any piece of text. It can be used to assign the subject categories, topics, spam detection in email, authorship identification, Age/gender identification, language identification, sentiment analysis, etc.

2.4.1 Text Classification Process

Bhumika, Prof Sukhjit Singh Sehra, Prof Anand NayyarText, they define the Text Classification Process and explained those steps. This classification Process consists 6 major parts, Documents, Pre-Processing, Indexing, Feature selection (Reduce dimensionality), Classification algorithms and Performance measure [19].

- **Documents**

In this step, we are collecting the documents which are different formats, such as html, pdf, and doc.

- **Pre-Processing**

This Process step, we are convert the documents into suitable format (clear format). typically in this step consists 3 major parts, Tokenizing, Removing stop words and Stemming words.

- **Indexing**

Vector Space Model is one of most commonly used document representation method in Natural Language Processing. In this model documents are represented by vector of the words. In this step, reduce the complexity of the document and make it easy to handle, the document convert from text document into vector document. Indexing led to some serious issues, such as high dimensionality of the representation, loss with adjacent words loss correlation, loss semantic relationship that exist among the term in a document.

- **Feature selection**

To improve the efficiency, scalability and accuracy of the text classifier, we have to do feature selection. The main thing in feature Selection (FS) is we have to select subset of features from particular original document. By applying this selection method, we can reduce the dimensionality of feature space, can improve the performance of classification and computing complexity will reduce. There are many feature selection methods notable in text classification, such as Document Frequency (DF), Mutual Information (MI), Information Gain (IG), Gini Index, Chi-square, expected cross entropy [23].

2.4.2 Methods and Techniques for Text classification

There are number of classification methods and technique based on machine learning can be applied to Text classification. Some of them are Naïve Bayes Text Classification, Support Vector Machine (SVM), Decision Trees, regression models, Neural Network, K-Nearest Neighbor.

Naïve Bayes Text Classification, Vector Machine (SVM) and Neural Network are commonly used for text classification purposes.

- **Naïve Bayes Text Classification**

Naïve Bayes Text Classification is one of the best classifier in text classification, and this is based on Naïve Bayes Theorem.

When Bayes rule applied to a particular document (m) and a class c,

$$p(c|m)=p(m|c).p(c)/p(m).$$

Here $p(c|m)$ is posterior probability of c given m, $p(m|c)$ is prior , $p(c)$ is likelihood and $p(m)$ is evidence. If the training data is not perfect, in naïve bayes text classification we can't expect learning as desired. The great advantage of using naïve bayes text classifier is, it's learning time is very low [9].

- **Support Vector Machine (SVM)**

Support Vector Machine is one of the good text classification methods, which is based on vector space. Main purpose of this method is, how to draw a straight line, and has widest width, that separate two different kind of data points (two different classes).commonly SVM performs better than other notable classifiers.

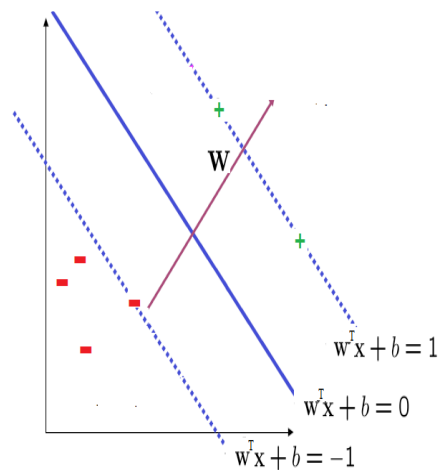


Figure 2.4: SVM classification methods [27]

- **K-Nearest Neighbor (KNN)**

In K-Nearest Neighbor (KNN), when want to classify the unknown instance vector “N”, KNN algorithms identifies the K nearest neighbors among training instance vectors of input unknown instance vector. Input unknown Instance vector assign to the class that most common neighbors contain class among k neighbors. Determine the Neighbors based on the similarity between input instance vector “N” and training instance vectors. Similarity measure by using Euclidean distance or the cosine between those two vectors. KNN is a lazy learning method.

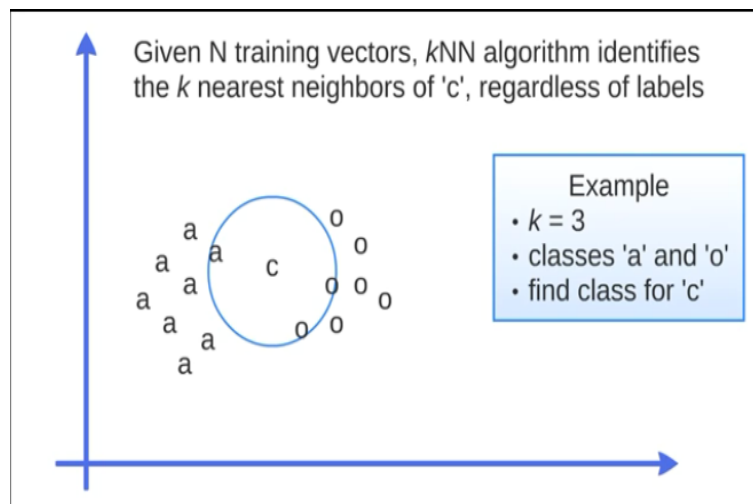


Figure 2.5: KNN classification methods [27]

2.4.3 Challenges in Text Classification

In this section, we discussed about the challenges in text classification based on other's work. Even it useful for topic categorize to any kind of text, when it automate using computer related components it is very complex task and lot of challenges. The major challenges in text classification are discussed below based on others works.

- **Word Sense Disambiguation (WSD)**

WSD is one of major problem in Natural Language Processing as well as Text classification. This problem discussed in various research papers, Alok Ranjan Pal and Diganta Saha, they discussed about WSD in their paper WORD SENSE DISAMBIGUATION.

And they mentioned the how WSD is related to Text classification. WSD helps to identifying the meaning of the word in a particular sentence when that particular word bearing more than one meaning. Word that bearing multiple meaning we call it polysemous word .Usually intelligent human can solve this problem easily and resolve the meaning of polysemous words. But when it comes to machine, it can't resolve the meaning of the polysemous words and will problem arise. Nowadays, Machine Readable Dictionaries (MRD) and Large Scale Corpus are growing quickly. Those things help to WSD related researches. WSD algorithms are categorized based on three types, Supervised, Unsupervised and Knowledge-based.WSD algorithms works based on the knowledge, that derives the knowledge from large scale corpus and Machine Readable Dictionaries (MRD).And also WSD algorithms is supervised or unsupervised depending on using of training data or not. Today, supervised learning methods are mainstream approach to WSD. Unsupervised WSD methods are usually do not assign the meaning to polysemous words instead those algorithms discriminate the particular polysemous word meaning based on information found in un-annotated corpora [17]. For this process these algorithms do not need external knowledge or Machine Readable dictionaries .And this kind of algorithms use for clustering purposes.

- **High dimensionality of the feature space**

High dimensionality of the feature space is another major problem in Text classification. Hyunsoo Kim Peg Howland and Haesun Park, they discussed this problem with Support Vector Machines (SVM). In Natural Language Processing Tasks, a word consider as a feature of a document or sentence. When a document has thousands of words, In Naïve base approach each words consider as a vector with thousand dimensions. Reducing this feature space is one of the major problems in text classification. Reduce the high dimensionality of the feature space to low dimensionality of the feature space is a major challenge. Reduction of the feature space led to little accuracy and gain performance in the results. Feature section can be done as supervised with the help of human or as unsupervised without any help of the human.

- **Training data**

Training data Classify the text manually is very hard and lot of time take for this task. There are many corpora available in web, those corpora created for evaluating purposes. Such as evaluate different algorithms, different techniques. But we can't get 100% accuracy results by using those corpora. There are positive and negative corpus available in the web for analyzes the sentiment value of tweets. For unsupervised text classification, Yarowsky proposed a method [25].this method can used to generate the training data, but unfortunately this method not guaranteed to give 100% accuracy.

2.5 Developing an Ontology

The vision of Semantic Web is structuring the knowledge that is present in the current web so that it's machine understandable without human intervention. Ontologies are the important backbone technology for the semantic web.

An ontology is an explicit (meaning of all concepts must be defined), formal (machine understandable) specification of a shared (consensus about ontology) conceptualization (abstract model of domain, identified relevant concepts, relations). Ontologies can be represented via Classes, Relations and Instances. Classes are abstract groups, sets, or collections of objects and represent ontology concepts. Classes are characterized via attributes. Attributes are name-value pairs [3].

To facilitate querying and reasoning, extracted information required to be properly structured. We analyzed the requirement and availability of information for the domain of tourist places in Sri Lanka. Extracting information related to tourist places in Sri Lanka is the main objective of our project.

2.5.1 Ontologies for information extraction

The advantages of Ontologies is representing complex information properly and offering a principled structure within which to merge the information extracted from parallel texts. Structuring of ontologies has both benefits and costs. Structuring can be done or automatically manually [3].

The important benefit of an ontology structure is inferencing task which implements on any ontology structure is productively constrained to collect knowledge required by the given task only. There are costs associated with maintenance of the ontology and software system built using the ontology. Even there are few costs associated with ontologies, they provide several benefits [21]. Those benefits of ontologies are described below.

- Reuse and organization of knowledge

There are several standard ontologies for many domains. They can be easily reused in other ontology based applications. And Ontology can be a combination of existing ontologies. We can save money and time by reusing existing ontologies and components.

- Browsing or searching

Ontologies support for querying and reasoning in an application. And ontology can be used by an intelligent search engine which processes user queries.

- Interoperability

Interoperability refers the ability to work with other ontologies without special effort.

- Communication between systems, between humans and, between humans and systems

Ontologies are both human machine and understandable knowledge structures. Therefore ontologies can be used to ensure interoperability between computer programming system and between humans at data and processing level.

- Computational inference

Ontologies can enable computational inferences. Computational inferences are useful to derive implicit facts to enhance traditional browsing and retrieval technology automatically. And they are useful in providing an instrument to model domain knowledge independently of the underlying system implementation.

2.5.2 Ontology Development

To develop an ontology selecting an appropriate language and an editor, from the existing ontology languages and editors is a very important task. According to the requirements, it can be built from an existing ontology can be reused. After the identification of ontology classes by analyzing ITGS domain or reusing existing ontologies from e-Tourism or Travel ontology, relations corresponding to these classes should be aligned with ontology. After create Ontology, it needs to be populated after the design process.

Ontology population is augmenting the ontology with instances of concepts and properties. Ontology population process has two sequential phases. They are the Information Extraction phase and the Population phase.

2.5.2.1 Ontology languages

There are several ontology languages such as Extended Markup Language (XML), Resource Description Framework (RDF), Ontology Interchange Language (OIL), DARPA Agent Markup Language (DAML), Simple HTML Ontology Extension (SHOE) and Ontology Web Language (OWL). In order to develop an ontology, selecting a suitable language is very important. An ontology language must have several requirements such as Have compact syntax, be highly intuitive to humans, have well defined formal semantics, be able to represent human knowledge, include reasoning properties, have the potential for developing knowledge bases and have a proper link with existing web standards to ensure interoperability.

2.5.2.2 Ontology editors/tools

Ontology editors are used to assist modelling ontologies. There are several ontology editors/tools such as Protégé [24], DOME (Distributed Ontology Management Environment), SWOOP, Ontolingua and Altova Semantic Works and etc.

- Protégé

Protégé is a free, open source ontology editor, knowledge requisition system and also a framework for developing intelligent systems. It is written in java and used Swing for complex user interface creation.

- DOME

DOME stands for Distributed Ontology Management Environment. It supports effective management of ontologies.

- SWOOP

SWOOP is a hypermedia-based Featherweight OWL Ontology Editor.

- Ontolingua

Ontolingua is developed by OnTO knowledge project and implements the ontology construction process.

- Altova Semantic Works

Altova is an RDF document editor framework and ontology development IDE. It creates and edits RDF documents, RDF schemas and OWL ontologies.

Protégé is a free, open source tool for ontology editing and management, knowledge requisition system and also a framework for developing intelligent systems. Protégé is the most widely used domain independent tool and other reasons of selecting Protégé that is freely available, platform independent technology for developing and managing terminologies, ontologies and knowledge bases in a wide range of application domains [16].

2.5.2.3 Aligning relations

There are main four types and categories of ontologies as top level ontology, task ontology, domain ontology and application ontology. Among these categories in our project, we are focused on domain ontology.

While developing ontology for an application, identifying its underlying relations properly is also an important task. ‘Is a’, ‘part of’, ‘has part’ and ‘regulate’ are some examples for ontology relations.

2.5.2.4 Ontology Population Process

The goal of Populating ontology is the extraction and classification of instance of the concepts and relations which are defined in the ontology development. There is a large number of ontologies for e-tourism application, by combine the existing ontologies is used for populating process. Population ontology is composed by two sequential phases. They are Corpus Processing Phase and Ontology Population Phase [4].

- **Corpus Processing Phase**

The main objective of first phase is processing the text to obtain set of semantic annotations. The annotations are the candidates for instances of ontology. General Architecture for Text Engineering (GATE) tool plugins as A Nearly New Information Extraction (ANNIE) have been used for this phase implementations.

- **Ontology Population Phase**

To continue this phase, the entities have to be properly labeled. The main objective of this phase is to maximize the amount of properties and relationships associated with entities in the ontology. This phase determines whether those annotations represent entities, properties or relationships in the ontology.

Ontology Population phase is language and domain independent. And this phase is responsible for solving ambiguities generated by Corpus Processing phase such as overlaps of semantic annotations, having more than one semantic annotation by linguistic expression, mapping into more than one ontological entity by semantic annotation [12].

2.6 Aggregating ratings

To make a decision on certain attraction of tourist places in Sri Lanka, our system required to calculate an overall sentiment value for each selected tourist places attraction with respect to concepts defined in ontology. Aggregator implements the calculation of overall sentiment value using the ratings given by sentiment classifier [6].

Recommender and Rating systems based on user review items by aggregating users' ratings or review in on-line. Ratings of places are aggregated by using a weighted arithmetic mean. The reviews express ideas about the places, experience, activities, events, fees and lot of other useful information regarding an attraction. By reading extracted information from reviews an overall idea about a tourist place can be obtained.

In the phase of information extraction, aggregating ratings is similar to the process of understanding an overall conception stated in a set of reviews.

2.7 Summary

There are more tools and technologies to reach the goals. But each of them have some benefits and drawbacks. And we try to find the suitable one to obtain the benefits from those technologies which were used earlier.

CHAPTER 3

3.0 Technology Adapted

3.1 Introduction

The framework basically targeted the tourism in Sri Lanka. Thus we used the technology according to our individual parts. In the beginning of our project we went through some research about the tools and technologies which can give proper solution for the problem. In this chapter we described the technologies which are used for the main areas as Text retrieval, Natural Language Processing and Mapping to Ontology.

3.2 Used Technologies and concepts

We mentioned into three main sections to describe the used technologies and concepts as Text retrieval, Natural Language Processing and Mapping to Ontology.

3.2.1 Used Technologies for Text Retrieval

We planned to take the data from online web sources, which are tourist based websites and user reviews in those websites. There are many sites available in web which are related to tourist places. And many of those sites are encrypted. If a site is encrypted we can't get the data from it, no any method to extract the data from it. Since Java provides HTML parsers for getting the data from HTML content pages which are not encrypted. Such as Jsoup and Jaunt.

3.2.2 Used Technologies for NLP

To manipulate the Natural Language Processing task with General Architecture for Text Engineering (GATE) software infrastructure it is open source application. GATE about General Architecture for Text Engineering, worked with GATE processing resources and make them for our NLP task easier by creating GATE processing resource pipeline which is more suitable for our task.

We extracted information from the reviews and other relative web articles using GATE processing resources and other plugins of the GATE infrastructure as well. Developed our project based on java platform for its robustness, adoptable many plugins and learning purpose of that platform as well. Although GATE is java platform base application and it provide largest java supportive tools and framework altogether make it collaborative development environment

Major goal of the GATE inventors were make easy way to interchangeable information among the Language Engineering (LE) modules, and enhance the different language source written modules and evaluate the components of the certain LE [11].

GATE infrastructure given us three main subsystems for perform task wisely,

- GDM - GATE Document Manager
- CREOLE - Collection of REusable Objects in Language Engineering.
- GUI - GATE Graphical User Interface.

The GATE provide these sub division categories for support the Language Engineering processes in handy way, it provide tasks which is handy for work with GATE, GATE support many plugging related to Language Engineering processes, it provides data storage and manage how data exchange with in the certain processes and visualization of the text after NLP processes have been done [1]. The GDM (GATE Document Manager) applicable for text mining processes and manage the processing resources to focus the task ordered manner. Language Engineering module contain and manipulate set of resources that are supportive to embedded the processing resources within the CREOLE plugins [11]. User Interface of the GATE infrastructure capable for visualize accesses and manage the working flow as well.

GATE provides different platform for different end users, users may be a researcher or developer or a student following a tasks through the GATE infrastructure. As a developer GATE provides different level of plugging for conducting any suitable modules for make their task easy, ANNIE (A Nearly New Information Extraction)plugging is an attractive resource module, it consist many NLP tasks with in real beneficial processing resources.

And also GATE provides tools for access the data from the central databases, and also GATE focuses on evaluate and manage the data from the processing resources and capable for loading processing resource modules in dynamically needed areas. GATE provides to do tagging ability with sequentially for enabling and enhance the individual elements not ambiguously. Also it cares of without make any changes to other elements when consider an individual elements operations [1].

GATE considers domain independent languages processing modules, such as NER- Named Entity Recognition components it used in GATE infrastructure it may collaborate with common GATE specific Gazetteers. GATE supports the visualization modules and documentations, and also support reuse components as well. And this framework very keen to determines the Machine learning environment as providing some tools to generate domain knowledgeable training data and other plugins are available for Machine Learning purposes.

3.2.3 Used Technologies for Mapping to Ontology

Semantic Web and Ontology Modelling is used to refer to the formats, tools and technologies that enable it. The technologies as Resource Description Framework (RDF), notations as RDF Schema (RDFS) and Web Ontology Language (OWL), all these technologies are regarded to provide a formal description of concepts and relationships. We decided to model RDF graph by using the Java based framework called Jena. Jena framework is based on Java deals with programmatic statements.

Protégé tool had to make many design consolation. And Jena has been designed for RDF and OWL. In our research Protégé-OWL used Jena framework for parsing and provides a Jena “view” which is implementation of the Graph interface. So that some Jena functions can be used for Protége. From that it is very easy to create an ontology file with Protégé.

For the system implementation we analyzed the most common capabilities specific to Semantic web applications. For the efficacious results tools and techniques; Protégé IDE, Jena Fuseki server, Jena API, SPARQL, Spring MVC and JSP were used predominantly.

3.2.4 Used technologies for text classification

We use WEKA framework for the text classification purposes. WEKA is open source java package, it contains collection of Machine Learning Algorithms. We have used LibSVM, J48 and Naïve Bayes models as subject classifier. And linear regression and SMOreg (Regression for SVM)

3.3 Summary

In our research project we try to use various technologies to implement the elements of the final outputs. At the moment we are in starting stage. And the technologies may be vary at the end of the project while doing the project.

CHAPTER 4

4.0 Approach for ITGS

4.1 Introduction

In this chapter we described the approach which was taken to reach the goals of the system. And furthermore described the technologies which are used up to now in our system.

4.2 Text Retrieval Approach

We planned to take the data from online web sources, which are tourist based websites and user reviews in those websites. And also we have planned to get the data from Social Medias, such as Facebook, Tweeter, Instagram and YouTube comments. In Facebook we planned to get comments and posts based on the hashtags. But we could get more reviews from tourist based websites. Therefore decide to get reviews form only websites .Most of the websites are written in HTML, some of those sites are encrypted. We don't have any method to extract the data from encrypted pages, since java provides some libraries for getting the data from HTML content pages which are not encrypted such as jsoup and jaunt.

We decide to extract the data content from web using jsoup library and jaunt library. Jsoup is only working with real-world HTML content. Jsoup provides a suitable API for extracting data. Java provides a library “restfb” for getting the data from Facebook. And get the data from Facebook we need to get a life time access token from Facebook, we can get the lifetime access token from “developers.facebook.com”. Also Java provides a library twitter4j for getting data from twitter, get the data from twitter we need to get the Consumer Key (API Key),Consumer Secret (API Secret),Access Token and Access Token Secret from twitter, we can get these items from apps.twitter.com

4.3 Language Engineering Approach

Initially data extracted from the web contents as online reviews, social media reviews and comments as tweeter and other Medias well. All the selected reviews are stored in data storage which is provided by GATE. Then stored data annotated regarding to configurations of GATE pipeline processing resources. Then phrases of the text representations were identified as subjective phrases and sentimental phrases separately.

And next separated subject phrases sent to classifiers and map relevant ontology model classes, for subject phrases can predicted by subject classifiers. And also sentiment phrase sent to the sentiment classifiers, for sentiment polarities given by sentiment words and it can be anticipated as ratings. The following figure shown about our NLP approaches.

4.4 Semantic Web Approach

Semantic Web and Ontology Modelling is an explicit specification of a set of objects, concepts, and other entities or classes that are presumed to exist and the relationships. According to our research, we developed an ontology to create relationships between places and their attributes.

4.5 Text Classifier approaches

Machine learning techniques for the success, depends on huge data representations. For the predictive model feature engineering is the technique to provide accurate solutions for the feature predictions and better improvable data model [46]. Selected feature is more important to influence the accurate level of the results, therefore we need to more care about selecting relative features.

we mentioned below feature engineering techniques for subject classifier and sentiment classifier, we try to explain rest of this sections by through a review, whereas this scenario for getting more understand to what we mentioned in below sections.

The scenario- "Arugambe is the most of the prettiest view of the beach, a long expense of white sand, fishing boats and under the trees a few guys are selling cold drinks."

4.5.1 Subject Classifier

Subject classes defines in the ontology domain subject phrases, after getting subject separately by the Natural Language Processing tasks, as GATE annotated outputs send to the relevant classifiers. Subject classifiers map the ontology class subject phrases which is in the annotated subject phrases. We decided to train the classifier using some approaches, such as

- Feature- Matching string based

This feature is determine the simple subject phrases in the annotated document, normally it considered simple subjective phrases in the document. Considering the above scenario the feature should triggered “view of the beach”

- Feature - Matching gazetteer based

This is depends on gazetteer phrases, subjective phrase need to match with gazetteer value. It can be used manually created domain knowledgeable gazetteers, most of the time it could be common long listed subjective values for in the gazetteer. According to our ontology classes, gazetteer list contain all of the subjective classes relative to the ontology classes.

This feature should triggered above scenario as Scenic category, the gazetteer contain the word “view” this should map with “Scenic” class of the relevant ontology class.

- Feature - Matching neighborhoods based

And we decided to enhance of our training data set by providing more approaches, This neighborhoods matching feature describes windowing and some kind of moving window techniques, window size we decided to three words it means there are three words in left and right side of the identified subject. Whereas window of words need to recognize the context through the subject based on particular window by its contained subjects.

And also this feature depends on gazetteer that used word list that belongs to words it contains window of the certain subject in the training data set. And it classify to relevant subjective classes in the identified window words, when in the classification the

matched neighborhood instances compared to the certain list of neighborhood gazetteer. Trigger out the maximum matching instances among the neighborhood instances and the neighborhood matching list of gazetteer.

4.5.2 Sentiment Classifier

We planned to develop following features for the sentiment classifier.

- Adjective based

Adjective based feature use for sentiment classifier, due to user expressed the review based using adjectives. We decide to get a gazetteer of adjectives, and planned to dive into 3 parts as positive, neutral and negative. When give an adjective words, we can't say that particular adjective is positive or negative.

For an example, deep is adjective and we can classify it different ways.

- Deep wave: negative
- Deep beach: positive

- Adverb based feature

Adverb is playing major role, when calculating the sentiment value for a word .as example most beautiful and beautiful are different sentiment value. Here “Most Beautiful” is more intense than just “Beautiful” and “Most Beautiful” is get the higher rating value.

4.6 Inputs and Outputs

The framework is able to extract reviews from World Wide Web and capable with the store extracted data into the databases. Then system will provide the aggregated rating of the places according to its procedure.

4.8 Summary

In this chapter was mentioned on the approaches used to achieve from the beginning of our research up to now. We have developed by using Java, Ontology, Protégé, GATE and few other technologies. According to the requirements these technologies may be changed later.

5.0 Design and Implementation

5.1 Introduction

This chapter includes the top level architecture of ITGS and processes among them. And Designs and Diagrams are mentioned with description.

5.2 Top Level Architecture of the system

The basic methodology of our project is to implement the web based intelligent system for tourism. The methodology followed in our system contains data retrieval, developing ontology, text classification and aggregating rates. The following figure shown abstract view of our main architecture.

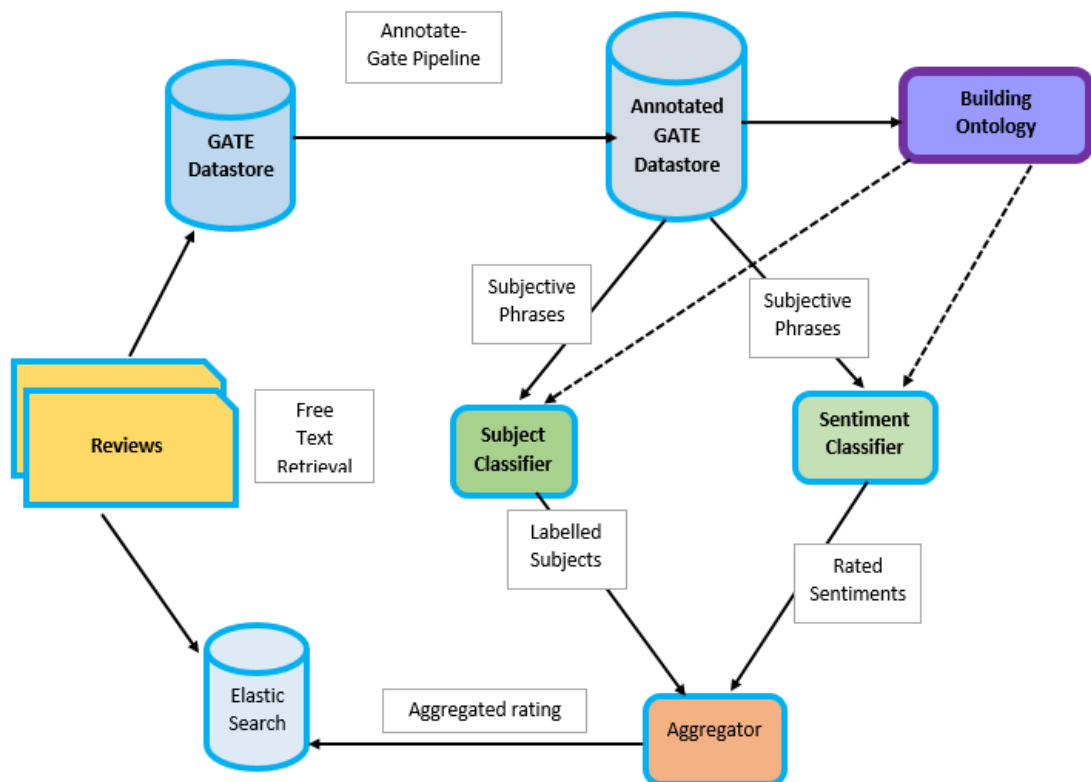


Figure 5.1: Top Level Architecture of ITGS

5.2.1 Annotated GATE Pipeline

We decided to Preprocessing data through GATE pipeline and then preprocessed data can be easy to classify as our classifiers. GATE given us preprocessing resources (PR) such as, Document Reset, Tokenizer, Sentence Splitter and Part of Speech Tagger and JAPE transducer. The following figure shown the working flow of the pipeline from ANNIE plugin to JAPE transducer

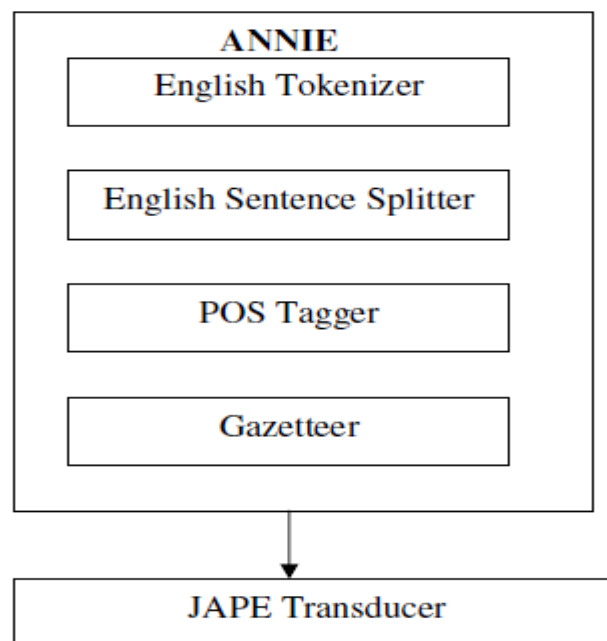


Figure 5.2: Annotated GATE Pipeline [22].

- Document Reset

This the first step for our defined pipeline processing resource. It can detect already existing annotation in the extracted data and clean or erase any previous set of text annotations, mainly removes HTML, wiki markups and other comments or tags as well [22].

- **ANNIE English Tokenizer**

In an extracted data set there are many different phrases/words in that document, we need to separate those words into tokens then it should added to Token set of annotation category. So the ANNIE English Tokenizer can provides above mentioned services for us. Text is splits into many categorizations as words, numbers, punctuations, space characters and so by the tokenizer. For the reason of the tokenizer is

- Efficiency maximization and enhance the flexibility of text mining processes.
- And which is make work easier for further developments of grammar rules [8].

The following figure shown Document with tokenizer in GATE GUI.

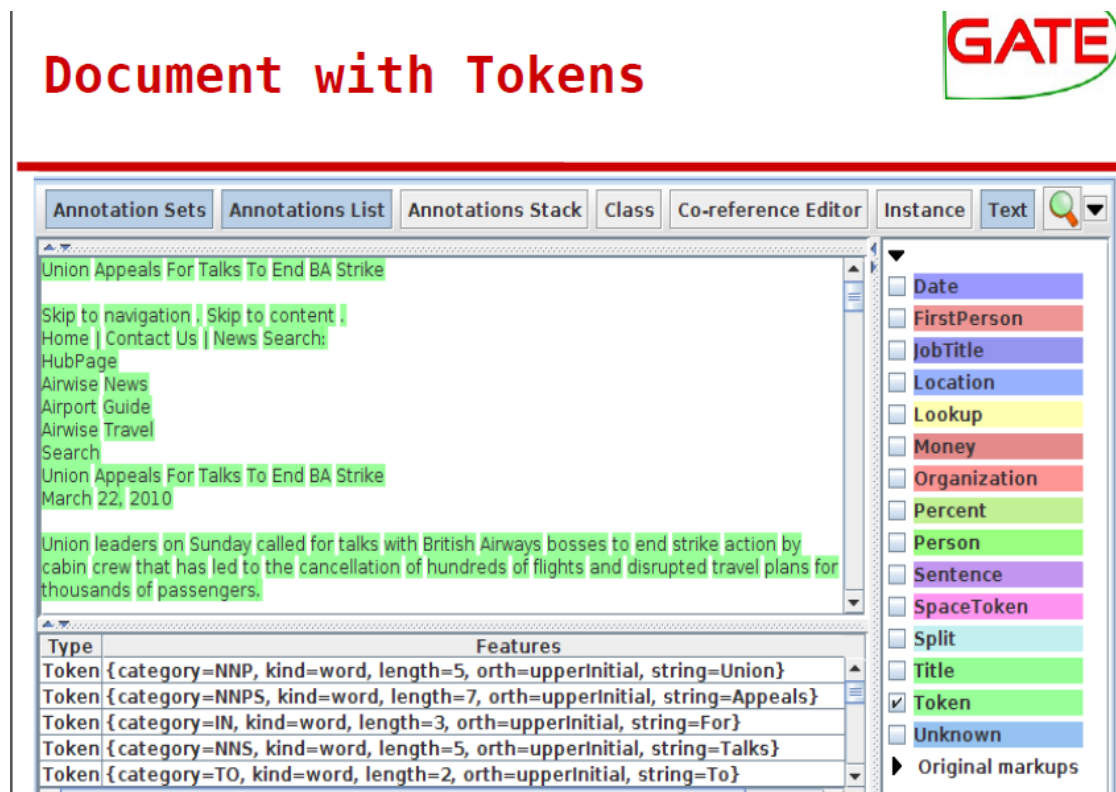


Figure 5.3: Document with tokenizer in GATE GUI [22]

- **ANNIE Sentence Splitter**

Extracted document may have several lines pages, so need to splits into sentences for added annotation on sentence annotation then it will separately identified sentences in a large text set of documents. It main purpose is segment the extracted text into sentence. This ANNIE splitter used gazetteers it may common domain knowledge level gazetteer [22].

The following figure shown Document with sentences in GATE GUI.

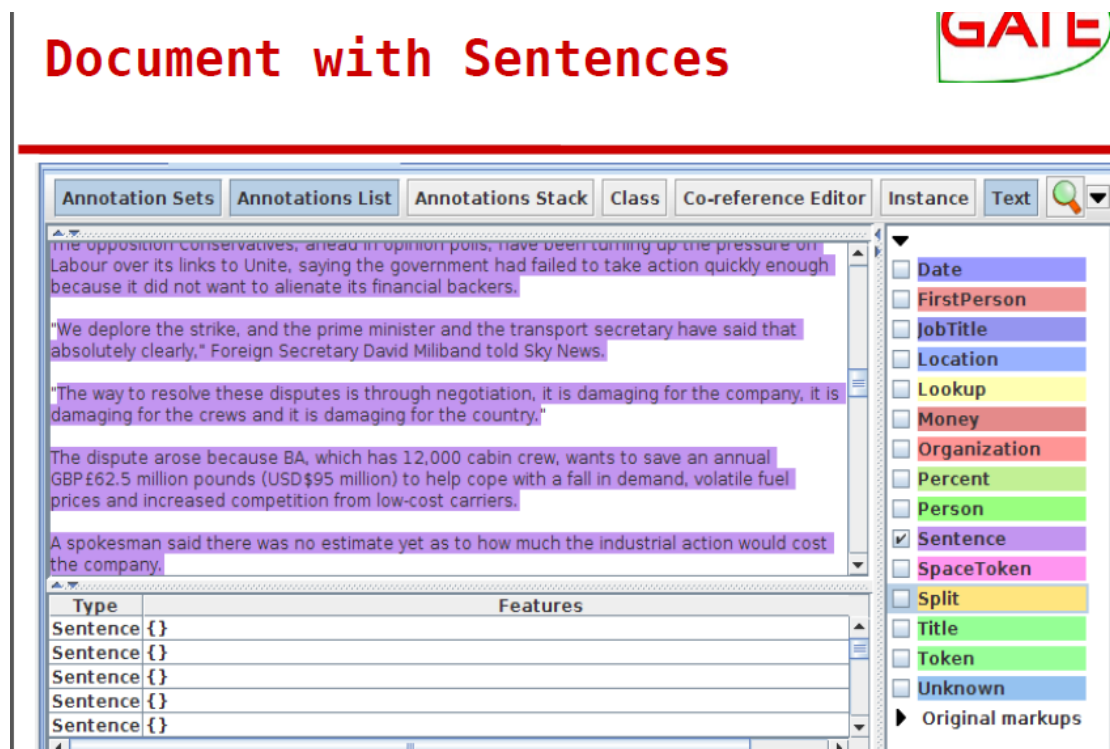


Figure 5.4: Document with sentences in GATE GUI [22].

- **ANNIE POS Tagger (Parts of Speech Tagger)**

It requires both earlier mentioned processing resources for the proper processes of POS. This can provides a part-of-speech tag as text document an annotation within each phrase or word and symbols. This determines tokens which is labelled as verb, noun and pronouns in the document. It may use a default set of rules according to the certain lexicons.

- **ANNIE Gazetteer**

Gazetteer has more variety of words, words in the gazetteers mapped by annotation type and certain gazetteer. Gazetteer resource have set of parameterized entities such as major type values and minor type values, and mainly it uses for recognizes the entity names of text based on gazetteer list. Commonly gazetteer lists are plain text list files it contains one line representation for an entity, such as Cities names, Organizational names, Days and other natural name entities there, and we can manually add our own lists if it is needed [22].

- **JAPE Transducer**

JAPE (Java Annotation Pattern Engineering) is rule base approaches, which is used to match the patterns in jape files and the document. It provides state of transductions over entire specific annotations based on the regular specified expressions. It is mainly determines the pattern matching languages for GATE infrastructure.

Jape rule consist two phases right and left rule phases.

- LHS - for it consist of pattern match rules.
- RHS- for it tell the details which type of annotation to be created when the RHS is triggered.

The following figure is an example for format of a Jape rule.

```

Rule name
Rule: University1

LHS
(
  {Token.string == "University"}
  {Token.string == "of"}
  {Lookup.minorType == city}
)
:orgName

-- >

RHS
:orgName.Organisation =
  {kind = "university", rule = "University1"}

```

Figure 5.5: example of format of a Jape rule [22].

And the following figure expresses about a review in a GATE document is executed in the resource pipeline of GATE, which is above mentioned PR.

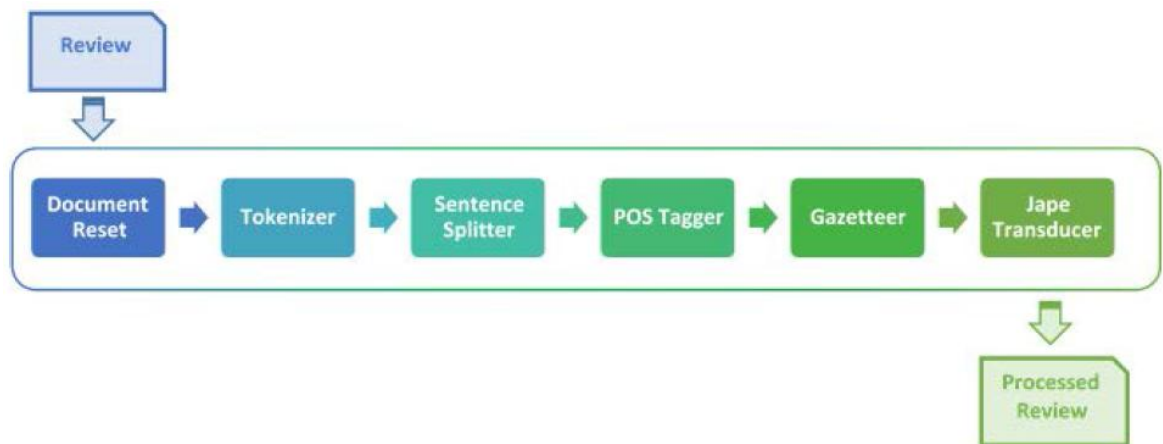


Figure 5.6: GATE pipeline [22]

5.2.2 Design of Text Classification

From the output of annotated GATE pipeline, to classifier will predict as subjective phrases, the extracted subjective phrases are provided into proper ontology classes and subject classifiers. The below following figure shown the phase of subject and sentiment classifier.

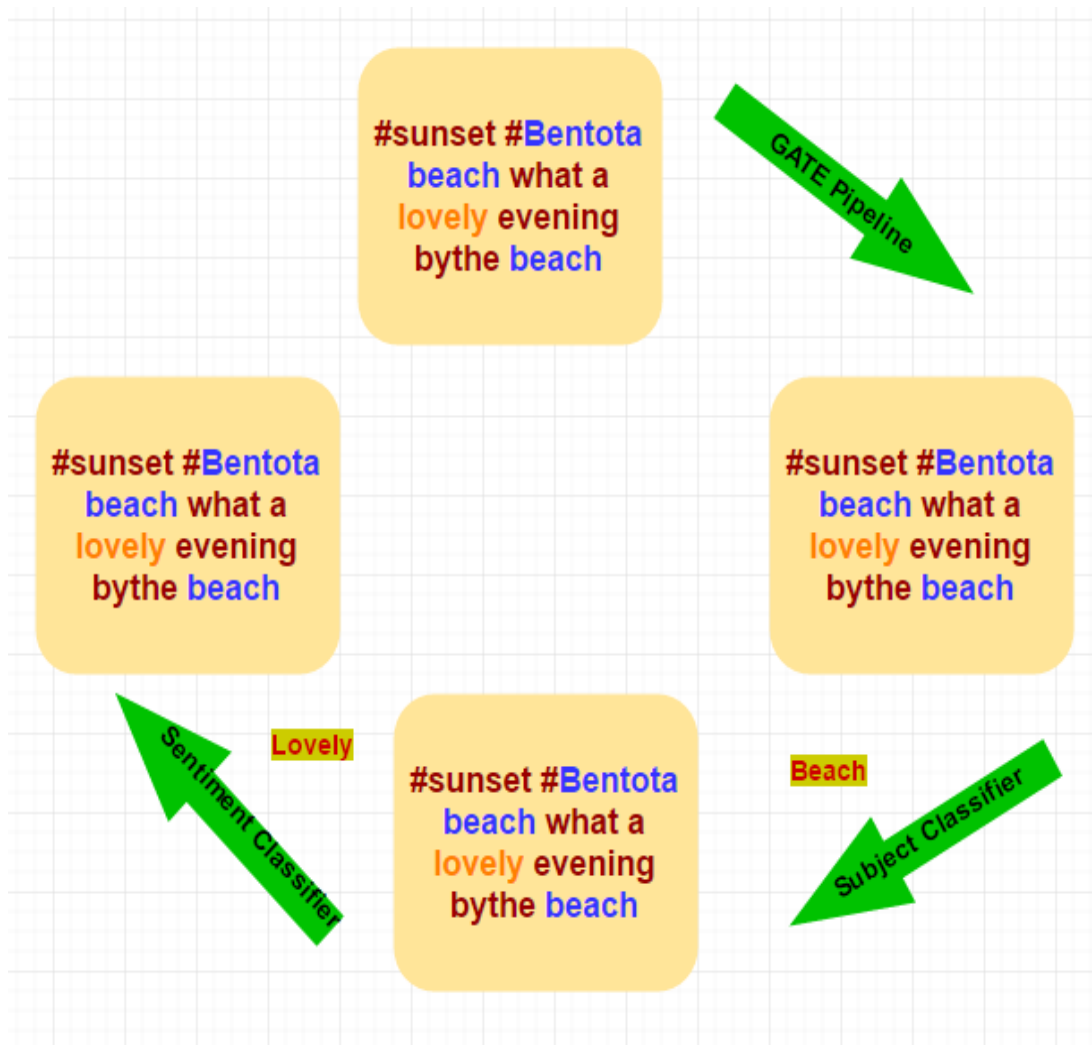


Figure 5.7: Annotations using GATE and Classifiers

We collect the reviews which are related to certain places, and we plan to analyze those tweets, how people are felt when sharing those tweets.

- Using Jsoup library we planned to collect the reviews from tourist based websites, about this task we discussed under the chapter 4 (Approach for ITGS),
- There are many unwanted things in a single review, we have to remove those things. Under the Pre-processing we discussed this task.
- Next step is we split the review into phrases.
- Identify the subject of split phrases under the subject classification using Naïve Bayes Algorithm.
- Next step is gives the sentiment value to those phrases under the Sentiment analysis using linear regression
- Finally calculate the overall of the sentiment value for particular place using mean approach.

5.2.3 Developing Ontology

There are 20 classes in the ontology designed for Intelligent Tour Guide System-ITGS.

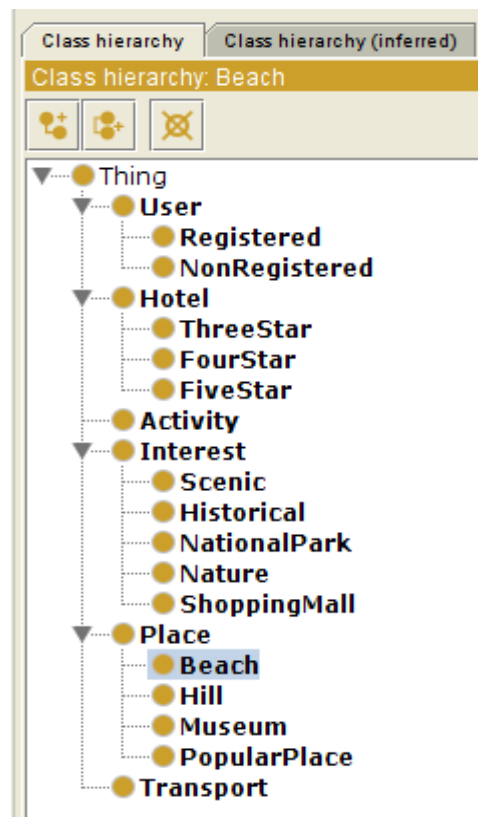


Figure 5.8: Ontology class hierarchy

5.2.3.1 Mapping keyword to Ontology class

Initial task was to map a relationship and implement the relationship using the Protégé IDE and save either in OWL format. With the stored dataset and SPARQL queries, we came across a method to access the dataset manually via Jena API along with use of java. OWL file can be stored locally or in a web space. The result set will be formed as ArrayList. Then the ArrayList object was split and taken into a result set and was sent to the JSP front-end as an inner html response.

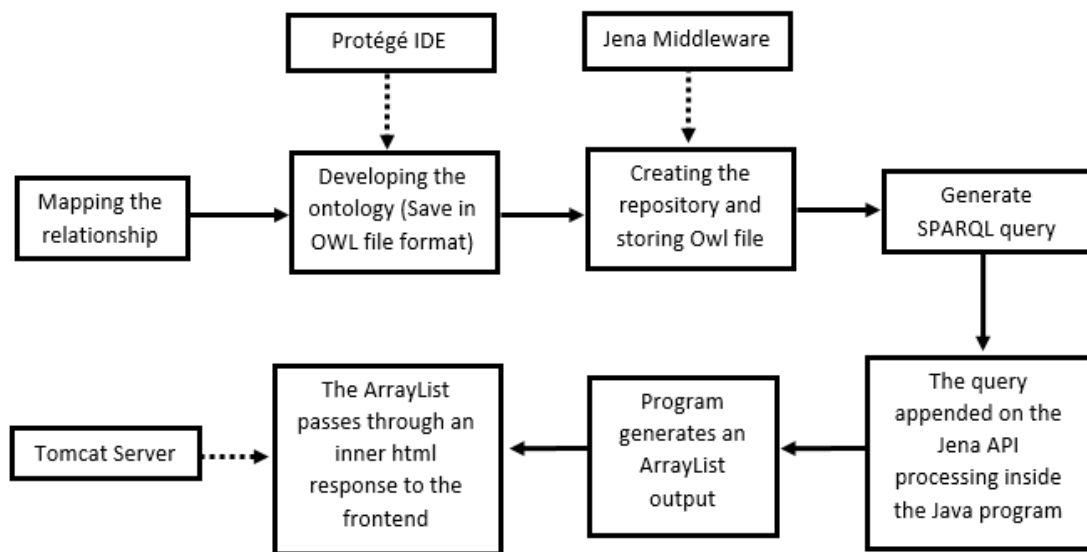


Figure 5.9: Process flow of implementation of semantic web in ITGS

5.2.4 Aggregation rating

There are several number of reviews that having information about places and attractions in Sri Lanka. Therefore to take a proper decision on a tourist places, from that integrate the sentiment value provided by obtained reviews need be measured.

Sentiment value of reviews is measured by sentiment classifier according to the phrases of subject. To analyze aggregated ratings to relevant reviews and places with respect to the entities defined in ontology modelling, rating aggregator need to be implemented. The following figure shown the sample processes of rating aggregator.

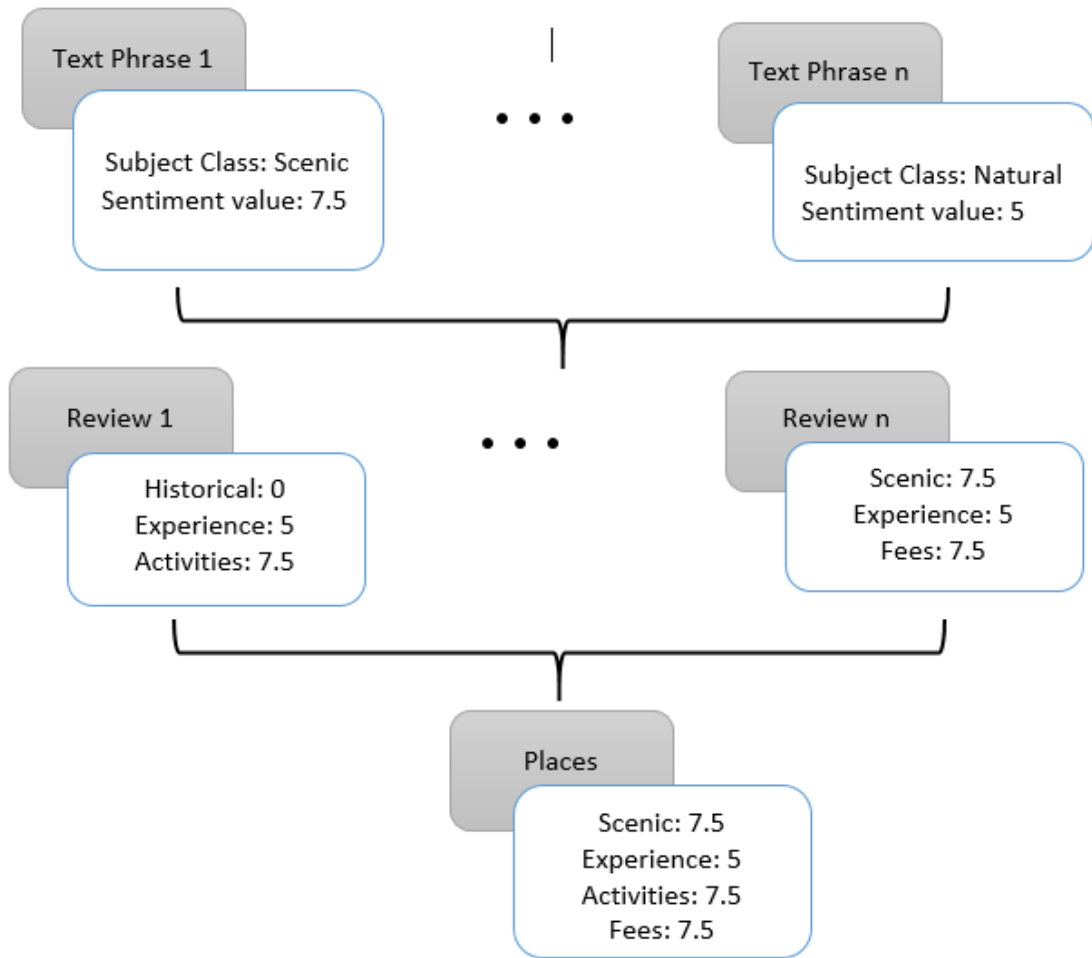


Figure 5.10: Aggregation rating for reviews

We planned to implement the aggregation rating based on median and mean methods, but mean is low sensitive to outliers and biases, and it is not suite for accurate aggregating rating [28]. Compared to mean approach, median is high sensitive to outliers and biases and it is most recommendable approach to aggregated rating

Using the variance we can get the biased values of a data set, and variance can calculate how far each number in the set is from the mean. With the purpose that, we decide to assign a threshold value for variance by detecting the possible rating and outliers. When the variance exceeds the threshold value, we defined that, particular data set includes outliers. If the data set have high variance, that means greater than threshold value, we planned to use median approach for aggregated rating, and if the data set does not have outliers, that means variance is less than threshold value, we planned to use mean approach for aggregated rating.

The first step of aggregating rating is calculating the sentiment value of each extracted phrase which belongs to define ontology class in each review. eg: in n^{th} review 1st phrase is belongs to nature class, sentiment value of that nature phrase is 5.0, likewise we calculate sentiment value of each phrases in each reviews. Next step is aggregate the sentiment value of each ontology class, and finally get the mean of aggregated sentiment value of each ontology class. e.g.: overall sentiment value for Scenic class 7.5. The means of each ontology class sentiment value is the overall attraction sentiment value for particular place. Above image shows overall process of text classification.

5.3 Summary

In this chapter was mentioned on design and some implementation from the beginning of our research up to now. We have designed text retrieval, text mining, developing ontology, text classification and aggregation rating module.

6.0 Experiment

6.1 Subject Classification

Selected Naïve Bayes classification as our subject classifier, by considered many experiment among J48 decision tree and LibSVM algorithms. This is considered to be a betterment of text classifier based on Naïve Bayes Theorem of probability. When the training data is not perfect, learning may not be as desired in Naïve Bayes Text Classification. The best thing about Naive Bayes it not worry about the Text Classifier missing data and also is that it takes very low Learning time.

The Accuracy of the Classification basically depends on size of the training data set, we commonly considered accuracy measures of the classification are

- Precision
- Recall

6.1.1 Precision and recall

"The fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that have been retrieved over total relevant instances in the data. Both precision and recall are therefore based on an understanding and measure of relevance. [7]"

- $\text{Precision} = \text{tp} / (\text{tp} + \text{fp})$ (tp - true-positive, fp- false positive, fn- false-negative)
- $\text{Recall} = \text{tp} / (\text{tp} + \text{fn})$

We consider the precision rate and recall by changing the training data set according to the certain results. And f- measure considers both the results of recall and precision for identifying the f-measure value and we can get an idea about the accuracy corresponding to the experiment via changing the data set as a variable for the classification model.

6.1.2 Training Data set

The training Data set we used for as a variable in our experiment to classifier was created using the phrases that were extracted from reviews of a website We used more than 500 manually labeled features for each and every evaluation technique for the subject classifier.

Used 10-fold cross- set for validation method to check the accuracy of the classification model and it provide to make proper data distribution of our testing and training data sets. Observing domain of travel and the neediness of the classification model, we get concluded f-measure, recall and precision as the most necessary measure for our subject classifier.

6.2 Predicting Sentiment Value

To improve accuracy of the Linier regression model, we increased the training data and get relevant Mean Squared Error for that particular data set. Therefore we manually create the training data by using collected reviews and we increased total number of training data and get relevant Mean Squared Error. We did this experiment for by increasing the training data.

Mean Squared Error defined as $\frac{1}{2m} \sum_{i=1}^m (h(X_i) - Y_i)^2$ here $h(X_i)$ is predicated value and Y_i is actual value.

Actual value,	Predicted value
0.9773555548504418,	0.9766874972427912
0.9847319278346618,	0.9849377425159254
0.9615239476408232,	0.9615440141038323
0.9801960588196068,	0.9803657525192013
0.9847319278346618,	0.9849948925050995
0.9899494936611665,	0.9881891535051941
0.8164965809277261,	0.8169349070993206
0.9847319278346618,	0.984574721867338
0.9847319278346618,	0.984999644332801
0.9801960588196068,	0.9805070001160787

Figure 6.1: Actual value and Predicted value in linear regression

Above image show that, we get the results of Actual value and related predicated value using Linear Regression. When we increase our data, the error is decreased, that means Mean Squared Error is decreasing with training data. Therefore we came to a conclusion as accuracy of the linear regression is increasing with amount of the training data.

We collect data (reviews) from different websites using jsoup library and splits into phrases and we labeled the data by assigning subjects for each instance, this training data set we used for subject classification. Same as we assigned the sentiment value for each phrases and prepare the training data sets for sentiment analysis.

For Example we prepared the training data set for Beach following method,

@data

'Walk for miles', 8

'Good surfing', 8

Here, 'walk for miles' is a phrase and “8” is sentiment value for that particular phrase. Likewise good surfing is a phrase and 8 is sentiment value for that particular phrase.

7.0 Results and Discussion

7.1 Results

7.1.1 Subject Classification

We obtained the results from our experiment for subject classifier are shown below in graph. Thus we can concluded as with the increase of training (data set) instances, there is an overall improvement about the accuracy of subject classifier outcomes. Therefore having wide range of training data set will provide higher accurate outcomes as classified for the classification model.

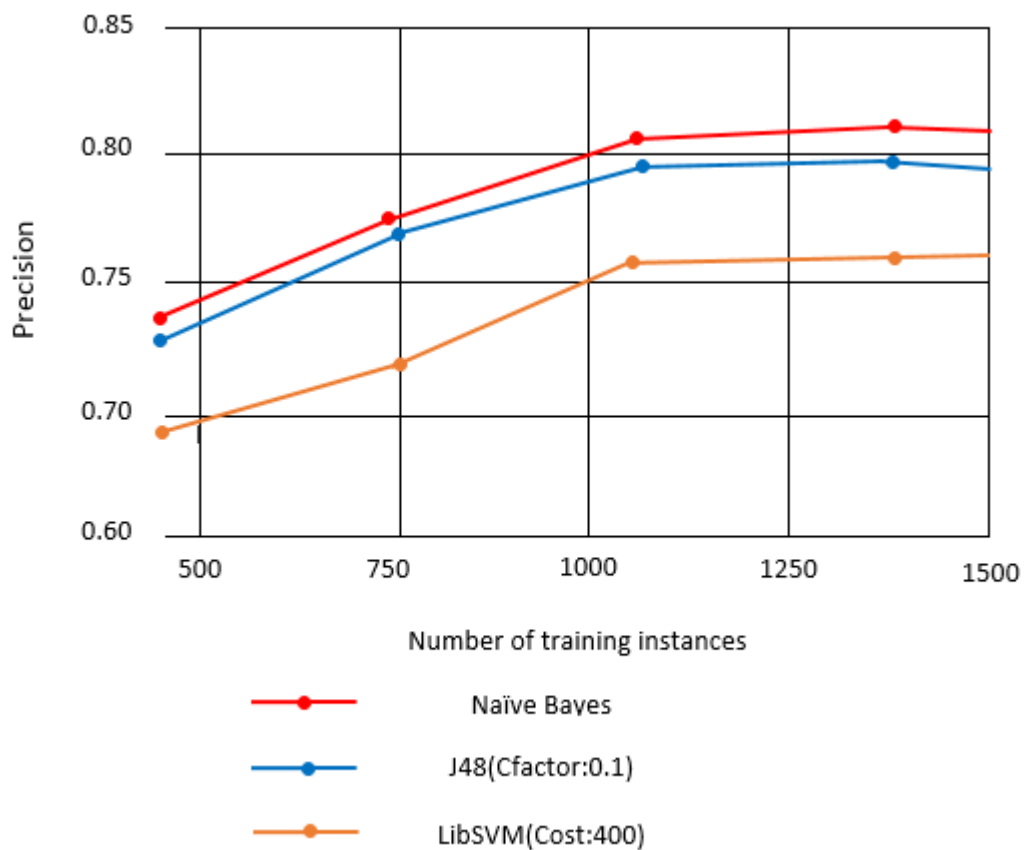


Figure 7.1: Precision vs training instances

As mentioned above classifier models, LibSVM classifier given poor results compared to other classifier model Naive Bayes. Therefore as the conclusion we selected Naive Bayes model for the implementation of our subject classifier.

7.1.2 Predicting Sentiment Value

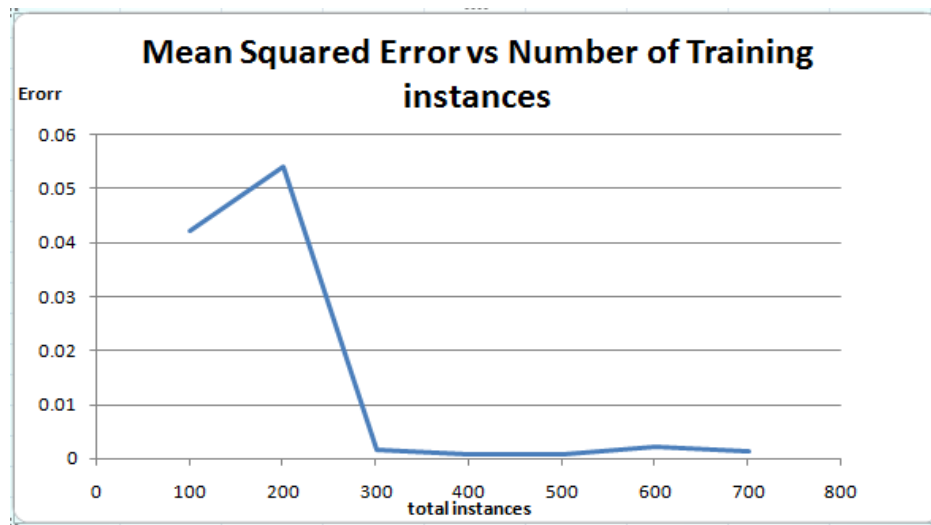


Figure 7.2: Mean Squared Errors vs Training Instance

Above graph is show that how means squared error change when training instance is increased. When training instance is 100 we get 0.042 means squared error, it decreased to 0.0018 when the total training instance is 300 and when total number of instance is 400 and 500 the mean squared error is decreased to 0.0009. But when the total number of instance is 600 and 700, the mean squared error is increase to 0.023 and after decreased to 0.0014. Therefore we can't say that the mean square error is decreasing with total number of instances.

Regularly over fitting problem arise in linear regression, to avoid this problem we have choose low order polynomial, so we can reduce the over fitting problem. Over fitting is we create the training data too well. When talk with linear regression, over fitting occur when polynomial curve goes to all data points in the training data set. so when a new data come, model will predicate wrongly and error will be very large, this means training error is zero and testing error is very large, to avoiding this problem we increased our training data instances. When we used the low polynomial curve, there is issue in that also, known as under fitting. In under fitting training error is very large and testing error is low. When polynomial curve is low, that particular curve is not going all data set in plan, but when a new data come to the plan, it will predict close to actual value. Therefore testing error will small.

7.2 Discussion

Text classification is a task which still does not have a unique method. The most naïve approach of predicting the class of the most occurring class of a given training set as the class of a given test set when there is no better answer, also gives a 60% correct classifier. This simple classifier for text classification is called “Baseline Classifier” and used as the benchmark for text classification experiments. Initially, in our project also we tried different approaches to achieve the accuracy of this baseline classifier.

In order to improve the subject classifier we decided to use a fine grained and coarse grained approach as discussed in section. In this approach, our idea was getting some idea regarding the prediction confidence of the classifier regarding the classification of instances. In case of Naïve Bayes and J48 Weka provided a float value of maximum prediction confidence for each classification it does. The value provided by J48 also gave some errors when it comes to implementation phase. However Naïve Bayes worked well but the accuracy of the classifier was not satisfactory comparing with the other two we focused on.

To get annotated output from the GATE tool, and we got the results and it further need to polish through some additional rules because of the language representation of the reviews more complex than we accepted, Whereas we designed more deeper rules in Jape transducer for get our needed output format accurately, and also there are bulky annotated results set in output data store even a single page of document also.

Text Classification is playing the major role in our project, due to text related tasks such as subject classification and sentiment classification. Weka framework is more suitable those tasks. We planned to use LibSVM, Naïve Bayes classification as subject classifiers and LibSVM and linear regression as sentiment classifiers.

Inference task that implements on any ontology developing is productively developed to obtain information. This is an important benefit of a developing ontology. Mostly the automated ontological structure systems reached their productivity benefits through the ontology editors. In addition to integrating for new ontologies, developing tools of ontology need to be customized with every ontology change. Due to these reason, there are costs associated with maintenance of the ontology and system built using the ontology modelling.

CHAPTER 8

8.0 Conclusion

Review for our research, was extracted from websites. Extracted wiki related reviews not enough for our train the classifier, and also there is no sentiment phrases on those reviews. So that we extracted data (user reviews) from only tourist based websites (Travel guider). For this task we have used jsoup library.

After getting bulky data from the online, need to do lot of preprocessing works before send to classifiers, thus preprocessed the reviews by eliminated unwanted links, correctors, labels and other cases. And develop GATE ANNIE plugging with JAPE rule for extract the features from reviews, for increase the accuracy level of the both subject and sentiment classifiers. Thus it can be given appropriate feature from the reviews which is most significant information in the certain reviews.

For implementation of subject classification, we used Naïve Bayes classifier. We compared the accuracy level of some classifier and choose one of them. We compared SVM (LibSVM), decision tree (J48) and Naïve bayes. Compared to other model, we get high accuracy in naïve bayes classifier. For implementation of sentiment analysis (calculate overall sentiment value) we used linear regression model. We compared the mean square error of SMO (SMOreg) and Linear Regression. And we get low mean square in linear Regression. Therefore we choose Naïve Bayes classifier for subject classification and Linear Regression for calculation overall sentiment value.

Ontology is a hierarchy of the most vital concepts related to a particular domain, their relationships and their properties. Ontology is an implicit, formal specification a shred conceptualization. The central notion of the Semantic web is to extend the current human readable web by encoding some of the semantics of resources in a machine process able form. For the Ontology development we analyzed the most common capabilities specific to Semantic web applications. For the efficacious results tools and techniques; Protégé IDE, Jena Fuseki server, Jena API, SPARQL queries were used predominantly.

We have introduced the process of implementation of Semantic web technology and the potential benefits to tourism. Converting data from the ontologies we created and extracting tourism related heterogeneous unstructured data from several web sites and using relevant tools and techniques, the web application was successfully implemented in Semantic web format. Implementation has been done using java.

In the on line, wide range of reviews can be found, which provide certain relevant information regarding Srilankan Attraction places. Therefore to get an idea about the tourist attractive places, the entire sentiment polarities expressed by the reviews of the relevant places need to be aggregated. Aggregator module developed for getting overall preferences of the certain interested places by through the reviews sentiment polarities. Developed the aggregator by considered mean of the sentiment values which created from the regression classifier.

It can be concluded that developed ITGS can identify the individual of the class by given key word according to relevant context, By undergo with set of predefined ontology classes and formulate the rate by observed the sentiment polarity of the given phrases and aggregate entire rates to create the aggregated rated values for each identified individuals according to given phrases.

References

- [1] "Apache UIMA," The Apache Software Foundation, 2013. [Online]. Available: <https://uima.apache.org/doc-uima-why.html>. [Accessed 11 2016].
- [2] Bird, Steven and Edward Loper. "NLTK: The Natural Language Toolkit". 1st ed. University of Pennsylvania: 2014.
- [3] S. Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text," Machine Learning, vol. 34, no. 1-3, pp. 233-272, 2009.
- [4] "Distributed Ontology Management Environment - User Guide," DERI Ontology Management Environment, [Online]. Available: <http://dome.sourceforge.net/guide.html>. [Accessed 11 09 2015].
- [5] E. Ferrara, P. D. Meo, G. Fiumara and R. Baumgartner, "Web data extraction, applications and techniques: A survey," Knowledge-Based Systems, 2014.
- [6] F. Garcin, B. Faltings, R. Jurca and N. Joswig, "Rating aggregation in collaborative filteringsystems", Proceedings of the third ACM conference on Recommender systems - RecSys '09, 2009.
- [7] Fiumara, Giacomo." Automated Information Extraction From Web". 2015. Survey. I-98166 Messina. Italy.
- [8] GATE user guide, 2016.[Online].Available <http://www.gate.ac.uk/sale/tao/split.html>. [Accessed 11 2016].
- [9] Goyal, Ram Dayal. "Knowledge Based Neural Network For Text Classification". 2009. Department of Computer Science and Engineering jalandhar.
- [10] Hart, Tabitha. "Introduction To TAMS Analyser For Mac OSX". 2011. University of Washington Seattle, WA 98195 USA.
- [11] H. Cunnningham, D. Maynard, K. Bontcheva, V. Tablan and Y. Wilks, "Experience of using GATE for NLP R&D," *Proceedings of the Workshop on Using Toolsets and Architectures To Build NLP Systems at COLING-2000*, pp. 1-8, 2000.

- [12] J. M. Ruiz-Martínez, J. A. Miñarro-Giménez, L. Guillén-Cárceles, D. Castellanos-Nieves, R. Valencia-García, F. García-Sánchez, J. T. Fernández-Breis and R. Martínez-Béjar, "Populating ontologies in the eTourism domain," Proceedings - 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 316-319, 2008.
- [13] Myllymaki, Jussi. "Effective Web Data Extraction With Standard XML Technologies". 2015. IBM Almaden Research Center. San Jose USA.
- [14] "Natural Language Toolkit," NLTK Project, 13 03 2015. [Online]. Available: <http://www.nltk.org/>. [Accessed 11 2016].
- [15] N. Ide and K. Suderman, "Bridging the gaps: interoperability for GrAF, GATE, and UIMA," in *ACL-IJCNLP '09 Proceedings of the Third Linguistic Annotation Workshop*, 2009.
- [16] "protégé," Protege.stanford.edu, 2015. [Online]. Available: <http://protege.stanford.edu/>. [Accessed 11 11 2016].
- [17] Ranjan Pal, Alok and Diganta Saha. " WORD SENSE DISAMBIGUATION". 2009. SURVEY. University of Kolkata.
- [18] Rehurek, Radim and Petr Sojka. Software Framework For Topic Modelling With Large Corpora. 2010.
- [19] Singh, Bhumika, Prof Sukhjot Singh Singh, and Prof Anand Nayyar." ALGORITHMS USED FOR TEXT CLASSIFICATION". 2012. REVIEW PAPER. Department of Computer Science and Engineering, Jalandhar.
- [20] Spivack, Nova. "Text Analytics Software With Sentiment Analysis, Categorization & Named Entity Extraction". Lexalytics.[Online]. Available: <https://semantria.com/>. [Accessed 11 2016].
- [21] T. Bürger and E. Simperl, "Measuring the Benefits of Ontologies," On the Move to Meaningful Internet Systems: OTM 2008 Workshops, pp. 584-594, 2008. Conference on Information and Knowledge Management (CIKM '10), pp. 9-18, 2010.

- [22] Thakker, Dhaval and PA Photos. "GATE JAPE Grammar Tutorial". Nottingham Trent University, 2011.
- [23] Vandana Korde et al Text classification and classifiers:” International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, March 2012”.
- [24] V. Jain and M. Singh, “Ontology Development and Query Retrieval using Protégé Tool,”International Journal of Intelligent Systems and Applications, vol. 5, no. 9, pp. 67-75, 2013.
- [25] Yarowsky, David. "UNSUPERVISED WORD SENSE DISAMBIGUATION RIVALING SUPERVISED METHODS". 2010
- [26] "ZKH Coder Index Page". Khc.sourceforge.net., 2016. [Online]. Available: www.khc.sourceforge.net/en/. [Accessed 12 2016].
- [27] “ZONE tutorials Text classifiers”. 2016. [Online]. Available: <https://www.youtube.com/watch?v=UqYde-LULfs>. [Accessed 12 2016].
- [28] F. Garcin, B. Faltings, R. Jurca and N. Joswig, "Rating aggregation in collaborative filtering systems", Proceedings of the third ACM conference on Recommender systems - RecSys '09, 2009.

Appendix A

Individuals Contribution to the Project

Index No: 124111R

Name: M. F. Mohamed

The main objective of our research project is extracting information related to attraction places of Sri Lanka from review data sources. Since my areas of interest is developing ontology and mapping key word to ontology class. Ontology modelling is a formal way of structuring knowledge of various domain. The extracted information should be structured properly to facilitate reasoning and querying. Thus we decided to use an ontology based approach for structuring information.

With the advice of our project supervisor, I have read several research papers, theses and articles on this topic. I have created a Tourism Ontology for ITGS using Protégé. Protégé is one of open source tool for ontology editing and management and freely available, widely used domain independent, platform independent technology for managing terminologies and developing. In order to develop an ontology, selecting a suitable language is very important, I have used Ontology Web Language (OWL) to develop a Travel owl file for our system. And according to research paper, to developing Semantic Web applications Jena which is a Java framework is more suitable one. Therefore the Jena services can be exploited for Protégé and it can be easy to develop an ontology with Protégé. After the design process ontology needs to be populated. I was completely new to Semantic Web and Ontological modelling, thus I had to spend lots of time on learning it.

For our system implementation we analyzed the most common capabilities specific to Semantic web applications. For the efficacious results tools and techniques; Protégé IDE, Jena Fuseki server, Jena API, SPARQL, Spring MVC and JSP were used predominantly. Then I have generated SPARQL queries by means of Apache Jena and ontology will be transverse according to specifications. We can retrieve the query result and passes the result to GUI or Front end. And we can delivers results to the front-end using Web services. Thence the user can input their requirements such as activities like hiking, swimming, shopping so on. Our system intelligently prepares a suitable journey accordingly.

We did ITGS- Intelligent tour guide system, this is a web based system for find the attraction places conveniently without wasting time with low cost. As a first step, we identify the major parts in our proposed solution, and we tried to dive the project into 3 scopes. Unfortunately we could not divide into three parts, and finally we divided as 5 parts. As my first task, I was assigned to “Free text retrieval” task. Textual information is playing major role in our proposed system (ITGS).getting all kind those textual information in a one place is not an easy task. First we limit the sources from which are we going to get the reviews and decided to get the reviews from tourism based websites and social medias, but we have got enough reviews from websites, so that we did not get reviews from social medias. Since there are many web pages are available in HTML format, and most of them are encrypted, and discussed this issue under the chapter 4. If the HTML page is not encrypted, we can get the data using web crawlers such as Jsoup, Jaunt. Therefore reviews from web pages (HTML based pages, which are not encrypted) were extracted using Jsoup library, and saved get reviews in MongoDB database

As my second task, Sentiment analysis under the Text classification. In this task calculate the overall sentiment value of selected tourist attracted places subject wise. Ex: for “Galle Face” Sentiment value for Activity: 8.5 and Sentiment value for Entertainment: 9.0. For this, I studied area of the text classification techniques, and regression techniques. An searched what are the common approaches for text classification. Those are Linear Regression ,Naïve bayes classifier, K-Nearest Neighbor (KNN), Support Vector Machine (SVM).Also identified challenges for the text classification task by go through the others work, these things are discussed under the others work. Weka” framework was used for the text classification tasks. Weka is a software, which contains collection of Machine learning algorithms, and written in Java. For calculating the sentiment value for subject labeled phrases, we used SMOReg (SVM) and Linear regression Library. After compare the mean square error of above mentioned regression models and we choose Linear Regression for calculating overall sentiment value of labeled phrase, and finally aggregate the overall sentiment value and get the attraction value for each subject class of selected places.

Index No: 124119A

Name: M. T. M. Nifras

We began the project with dividing the workloads in to three parts. In my interest of Artificial Intelligence arena, selected the NLP module and part of the Text classification module.

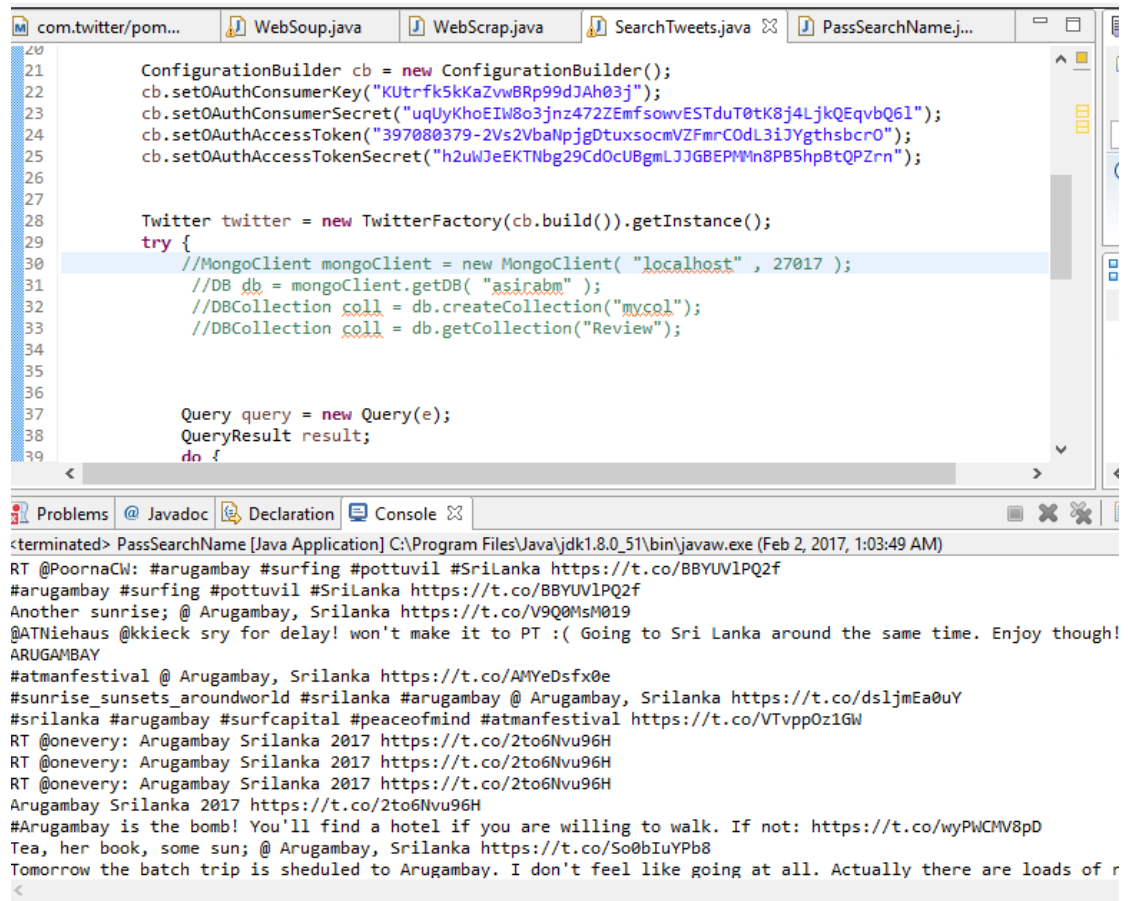
Worked on NLP module first, because of preprocessing is must needed part to the other Feature Engineering modules by myself. In the beginning stage I have not any knowledge about the NLP technologies and relevant tools, thus had to refer the relevant research papers and other documentation to get some ideas about the process. Compared the related works to capture advantages and disadvantages and selected most beneficial tool for preprocessing techniques. We prefer to worked with GATE because of it provide large scale of plugging environment of various resource processes and working on java platform. Decided to work through a pipeline of ANNIE plugging. Then worked to understand GATE infrastructure with beginning practiced in GUI, after that focused on embedded the framework in the working environment, whereas decided to create GATE CREOLE plugins in intellij idea. Faced some challenges in sort out the texts which only regard to our work thus created own JAPE rules for filtered to identified complex reviews/texts from the document set even though GATE provides processing resources for annotate the document, the actual work is not done because of complexity of the text representations of the different level people. Therefore added JAPE rules with help of other GATE processing resources accordingly.

The other module part is Subject classification, had to refer more research papers and relevant documents to get some understands about the Classification processes. And also compared about the some classifiers and classification tools among others approaches by the getting help with the research papers. After that we decided to implement the classification from the Weka tool. And used Naïve Bayes as subject classifiers by considered accuracy among the other classifier, despite we did some algorithmic approaches for feature engineering to classy the relevant subjective phrases and some experimental processes also considered to increase the accuracy of the classifier. Faced issues with the ARFF with the accuracy level according to our Classification problem, thus deal with some own algorithms to reduce the complexity of our own approach.

Some details of implementation

Free text retrieval

- Getting Tweeter reviews



```
20
21 ConfigurationBuilder cb = new ConfigurationBuilder();
22 cb.setOAuthConsumerKey("KUTrfk5kKaZvwBRp99dJA03j");
23 cb.setOAuthConsumerSecret("uqUyKhoEIw8o3jnz472ZEmfsowvESTduT0tK8j4LjkQEgqbQ61");
24 cb.setOAuthAccessToken("397080379-2Vs2VbaNpjdTuxsocmVZFmrCodL3iJYgthsbc0");
25 cb.setOAuthAccessTokenSecret("h2uWJeEKTNb29CdOcUBgmLJJGBEPmWn8PB5hpBtQPZrn");
26
27
28 Twitter twitter = new TwitterFactory(cb.build()).getInstance();
29 try {
30     //MongoClient mongoClient = new MongoClient("localhost", 27017);
31     //DB db = mongoClient.getDB("arugambay");
32     //DBCollection coll = db.createCollection("mycol");
33     //DBCollection coll = db.getCollection("Review");
34
35
36
37     Query query = new Query(e);
38     QueryResult result;
39     do {
```

Problems @ Javadoc Declaration Console

<terminated> PassSearchName [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (Feb 2, 2017, 1:03:49 AM)

RT @PoornaCW: #arugambay #surfing #pottuvil #Srilanka <https://t.co/BBYUV1PQ2f>
#arugambay #surfing #pottuvil #Srilanka <https://t.co/BBYUV1PQ2f>
Another sunrise; @ Arugambay, Srilanka <https://t.co/V9Q0MsM019>
@ATNiehaus @kkieck sry for delay! won't make it to PT :(Going to Sri Lanka around the same time. Enjoy though!
ARUGAMBAY
#atmanfestival @ Arugambay, Srilanka <https://t.co/AMYeDsfX0e>
#sunrise_sunset_aroundworld #srilanka #arugambay @ Arugambay, Srilanka <https://t.co/dsljmEa0uY>
#srilanka #arugambay #surfcapital #peaceofmind #atmanfestival <https://t.co/VTvpp0z1Gw>
RT @onevery: Arugambay Srilanka 2017 <https://t.co/2to6Nvu96H>
RT @onevery: Arugambay Srilanka 2017 <https://t.co/2to6Nvu96H>
RT @onevery: Arugambay Srilanka 2017 <https://t.co/2to6Nvu96H>
Arugambay Srilanka 2017 <https://t.co/2to6Nvu96H>
#Arugambay is the bomb! You'll find a hotel if you are willing to walk. If not: <https://t.co/wyPNCMV8pD>
Tea, her book, some sun; @ Arugambay, Srilanka <https://t.co/So0bIuYPb8>
Tomorrow the batch trip is sheduled to Arugambay. I don't feel like going at all. Actually there are loads of r

- Getting reviews using Jsoup

```

1 package com.webscrap;
2
3 import java.io.IOException;
4
5 public class WebSoup {
6
7     public static void main(String[] args) throws IOException {
8         Document doc = Jsoup.connect("http://www.lonelyplanet.com/sri-lanka/the-east/arugam-bay/int
9         // Elements els=doc.select("p.copy--feature");
10        Elements itemprop = doc.select("p.copy--feature");
11
12        // System.out.println(els.size());
13        // String keywords = doc.select("meta[name=keywords]").first().attr("content");
14        System.out.println(itemprop);
15        // String description = doc.select("meta[name=description]").get(0).attr("content");
16        // System.out.println("Meta description : " + description);
17    }
18 }

```

terminated> WebSoup [Java Application] C:\Program Files\Java\jdk1.8.0_51\bin\javaw.exe (Feb 2, 2017, 1:50:13 AM)

it break that many regard as the best surf spot in the country. It's a tiny place, with a population of a few hundred, and everything is dot
uses, oceanside restaurants and a mellow, swing-another-day-in-a-hammock kind of vibe that's totally removed from the brash West coast beac

- mongoDB results

```

Select Command Prompt - mongo.exe

{
  "_id" : ObjectId("588d613af267a9099046dc1d"),
  "Name" : "Arugambay",
  "Review" : "RT @MaheckChahal: Today I will do absolutely nothing! A very good morning frm #arugambay #srilanka #
surf #morning #lazy #chilling #sun #bea..."
}
{
  "_id" : ObjectId("588d613af267a9099046dc1e"),
  "Name" : "Arugambay",
  "Review" : "Arugambay Beach Villas and Chalets https://t.co/gyYom78vW6 https://t.co/7mz2HAKqUB"
}
{
  "_id" : ObjectId("588d613af267a9099046dc1f"),
  "Name" : "Arugambay",
  "Review" : "RT @MaheckChahal: Today I will do absolutely nothing! A very good morning frm #arugambay #srilanka #
surf #morning #lazy #chilling #sun #bea..."
}
{
  "_id" : ObjectId("588d613af267a9099046dc20"),
  "Name" : "Arugambay",
  "Review" : "RT @MaheckChahal: Today I will do absolutely nothing! A very good morning frm #arugambay #srilanka #
surf #morning... https://t.co/1LZgqUfvXp"
}
{
  "_id" : ObjectId("588d613af267a9099046dc21"),
  "Name" : "Arugambay",
  "Review" : "RT @MaheckChahal: Today I will do absolutely nothing! A very good morning frm #arugambay #srilanka #
surf #morning #lazy #chilling #sun #bea..."
}
Type "it" for more
>

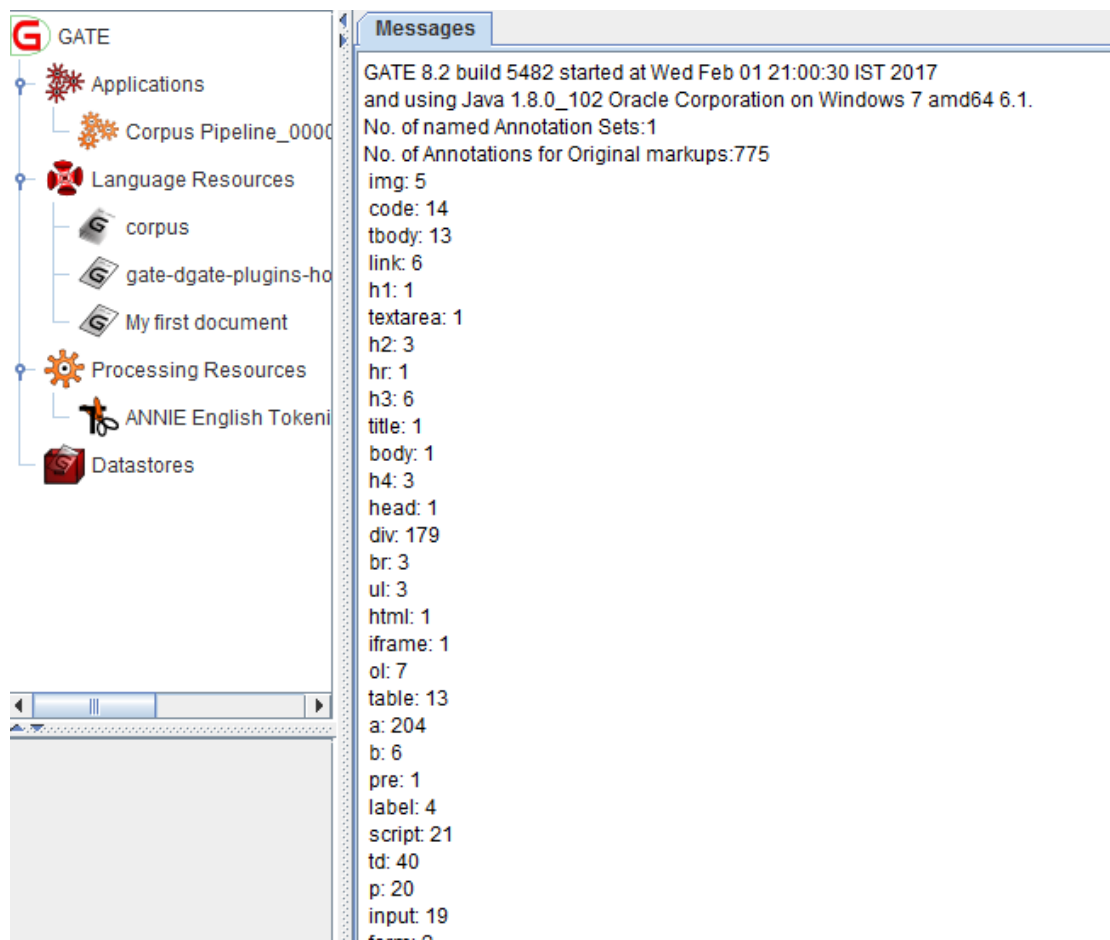
```

Text mining – NLP

- Screen shot - Embedded GATE to the working environment through the CREOLE plugins codes.

```
5 import gate.creole.SerialAnalyserController;
6 import gate.gui.MainFrame;
7 import gate.util.GateException;
8 import org.apache.log4j.Logger;
9 import org.apache.tools.ant.types.resources.comparators.Date;
10 import javax.swing.*;
11 import java.io.File;
12 import java.net.MalformedURLException;
13 import java.net.URL;
14 import java.util.Map;
15 import java.util.Set;
16
17 public class Main {
18
19     private static org.apache.log4j.Logger log = Logger.getLogger(Gate.class);
20     public static void main(String[] args) throws Exception{
21         if(Gate.getGateHome() == null)
22             Gate.setGateHome(new File("C:/Program Files/GATE_Developer_8.2"));
23         if(Gate.getPluginsHome() == null)
24             Gate.setPluginsHome(new File("C:/Program Files/GATE_Developer_8.2/plugins"));
25
26         Gate.init();
27         SwingUtilities.invokeLaterAndWait(new Runnable() {
28             public void run() {
29                 MainFrame.getInstance().setVisible(true);
30                 FeatureMap params = Factory.newFeatureMap();
31                 try {
32                     params.put(Document.DOCUMENT_URL_PARAMETER_NAME,
33                             new URL("http://stackoverflow.com/questions/28313692/gate-dgate-plugins-h
34                     params.put(Document.DOCUMENT_ENCODING_PARAMETER_NAME,
35                             "UTF-8");
36                 } catch (MalformedURLException e) {
37                     e.printStackTrace();
38                 }
39             }
40         });
41     }
42 }
```

- GATE GUI- text segmentation according to PR



- JAPE rule formation

```
1 Phase:firstpass
2 Input: Lookup
3 Options: control = brill
4
5 Rule: ScenicCategory
6 Priority: 20
7 (
8 {Lookup.majorType == "scene"}
9 ): label
10 -->
11 :label.Sport = {rule= "ScenicCategory" }
```

- Annotate the location if it is at the Beach

```
//rule identifies at the BEACH/Beach/beach
Rule: locationOfBeach
Priority:50
({Token.string == "at"})

(
    ({Token.string =~ "[Tt]he"})?
    (
        (
            {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
            ({Token.kind == punctuation, Token.subkind == dashpunct})?
            {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
        )
        |
        {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
    )
    (
        {Token.string =~ "[Bb]each"}
        |
        {Token.string =~ "BEACH"}
    )
):location
-->
:location.Location = {rule= "locationOfBeach" }
```

- Identify the location from the “at the” word

```
1 Phase: locationcontext2
2 Input: Lookup Token
3 Options: control = all debug = false
4 Rule: locationcontext2
5 Priority:50
6 ({Token.string == "at"})
7 (
8     ({Token.string =~ "[Tt]he"})?
9     (
10        (
11            {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
12            ({Token.kind == punctuation})?
13            {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
14            ({Token.kind == punctuation})?
15            {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
16        )
17        |
18        (
19            {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
20            ({Token.kind == punctuation})?
21            {Token.kind == word, Token.category == NNP, Token.orth == allCaps}
22            |
23            {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}}
24        |
25        {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}}
26    )
27    ({Token.string == ","})?((
28        {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
29        {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
30        {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}}
31    |
32    ( {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}
33    {Token.kind == word, Token.category == NNP, Token.orth == upperInitial}}
34    |
35    ({Token.kind == word, Token.category == NNP, Token.orth == upperInitial}}))
36 ):location -->
37 :location.Location = {rule= "locationcontext-locationcontext2" }
```

- Output of the above mentioned JAPE rules

```

23477 <Annotation Id="1579" Type="Sport" StartNode="84" EndNode="92">
23478 <Feature>
23479   <Name className="java.lang.String">rule</Name>
23480   <Value className="java.lang.String">CenicCategory</Value>
23481 </Feature>
23482 </Annotation>
23483 <Annotation Id="1580" Type="Sport" StartNode="84" EndNode="92">
23484 <Feature>
23485   <Name className="java.lang.String">rule</Name>
23486   <Value className="java.lang.String">CenicCategory</Value>
23487 </Feature>
23488 </Annotation>
23489 <Annotation Id="1581" Type="Sport" StartNode="84" EndNode="92">
23490 <Feature>
23491   <Name className="java.lang.String">rule</Name>
23492   <Value className="java.lang.String">CenicCategory</Value>
23493 </Feature>
23494 </Annotation>
23495 <Annotation Id="1582" Type="Sport" StartNode="84" EndNode="92">
23496 <Feature>
23497   <Name className="java.lang.String">rule</Name>
23498   <Value className="java.lang.String">CenicCategory</Value>
23499 </Feature>
23500 </Annotation>
23501 <Annotation Id="1583" Type="Sport" StartNode="84" EndNode="92">
23502 <Feature>
23503   <Name className="java.lang.String">rule</Name>
23504   <Value className="java.lang.String">CenicCategory</Value>
23505 </Feature>
23506 </Annotation>
23507 <Annotation Id="1584" Type="Sport" StartNode="84" EndNode="92">
23508 <Feature>
23509   <Name className="java.lang.String">rule</Name>
23510   <Value className="java.lang.String">CenicCategory</Value>

```

- Output of the above mentioned JAPE rules

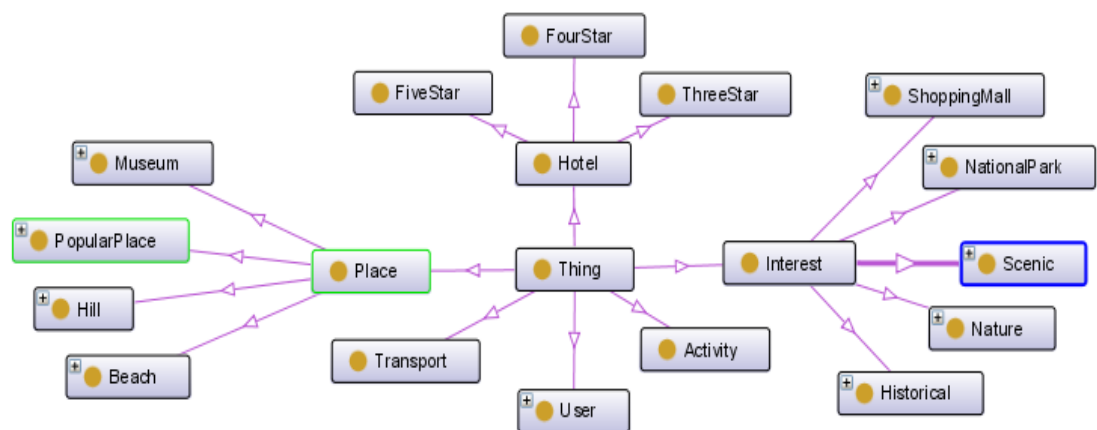
```

3  <!-- The document's features-->
4  <GateDocumentFeatures>
5  <Feature>
6      <Name className="java.lang.String">gate.SourceURL</Name>
7      <Value className="java.lang.String">file:/C:/Users,
8  </Feature>
9  <Feature>
10     <Name className="java.lang.String">MimeType</Name>
11     <Value className="java.lang.String">text/plain</Value>
12 </Feature>
13 <Feature>
14     <Name className="java.lang.String">docNewLineType</Name>
15     <Value className="java.lang.String">CRLF</Value>
16 </Feature>
17 </GateDocumentFeatures>
18 <!-- The document content area with serialized nodes
19
20 <TextWithNodes><Node id="0"/>Soccer<Node id="6"/> <Node id="41"/>Composite<Node id="50"/> <Node id="51"/>
21 <Node id="490"/></TextWithNodes>
22 <!-- The default annotation set -->
23
24 <AnnotationSet>
25 <Annotation Id="1" Type="Sentence" StartNode="0" EndNode="6">
26 </Annotation>
27 <Annotation Id="2" Type="Token" StartNode="0" EndNode="6">
28 </Annotation>
29 <Feature>
30     <Name className="java.lang.String">length</Name>
31     <Value className="java.lang.String">6</Value>
32 </Feature>
33 <Feature>
34     <Name className="java.lang.String">orth</Name>

```

Developing ontology

- Tourism OntoGraf



User Interface for Destinations Search

Search For Destinations

Rating Selected

Choose your option

☐ Parking

Activities

Choose your option

☐ Swimming Pool

☐ Fitness Room

SEARCH

Search For Destinations

Rating Selected

Choose your option

☐ Parking

Activities

☐ Choose your option

☐ Hiking

☐ Surfing

☐ Skiing

☐ Shopping

☐ Sight Seeing

Appendix C

Declaration

We declare that this thesis is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Students

Signature of Students

M.F. Mohamed

A.B.M. Asir

M.T.M. Nifras

Date: 03.06.2017

Supervised by

Name of Supervisor

Signature of Supervisor

Dr. (Mrs.) A Thushari Priyangika silva

Date: