# ABSTRACT

The ability to correctly identify the sentiment expressed in user-reviews about a particular product is an important task for several reasons. If there is a negative sentiment associated with a particular product, the manufacturer can take immediate actions to address the issue. Failing to detect a negative sentiment associated with a product might result in decreased sales. From the user point-of-view, in online stores where one cannot physically touch and evaluate a product as in a real-world store, the user opinions are the only available subjective descriptors of the product. Considering the numerous applications of sentiment classification such as opinion mining, opinion summarization, contextual advertising, and market analysis, it is not surprising that sentiment classification has received continuous attention. Sentiment classification can be considered as an instance of text classification where a given review must be classified into a pre-defined set of sentiment classes. In binary sentiment classification, a review must be classified into two classes depending on whether it expresses a positive or a negative sentiment towards an entity. Alternatively, a Review can be assigned a discrete sentiment score (e.g. from one to five stars) that indicates the degree of the positively or negativity of the sentiment.

# LIST OF FIGURES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIGURE NO.** |  | **NAME OF THE FIGURE** |  | **PAGE NO.** |
| 1.1 |  | Sentiment Analysis |  | 02 |
| 4.1 |  | System Architecture |  | 16 |
| 4.2 |  | Use case diagram |  | 19 |
| 4.3 |  | Sequence diagram |  | 20 |
| 4.4 |  | Class diagram |  | 21 |

# LIST OF ABBREVIATIONS

JDK Java Development Toolkit.

**JMF** Java Media Framework.

**TCP** Transmission Control Protocol.

**IP**  Internet Protocol.

**HTTP** Hyper Text Transfer Protocol

**CHAPTER TITLE PAGE NO**

**ABSTRACT** v

**LIST OF FIGURES** vi

**LIST OF ABBREVATIONS** vii

## 1. INTRODUCTION

1.1 SYSTEM OVERVIEW 1

1.2 SCOPE OF THE PROJECT 4

**2. LITERATURE SURVEY** 5

## 3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM 17

3.1.1 DISADVANTAGES OF EXISTING SYSTEM

* 1. PROPOSED SYSTEM 17

3.2.1 ADVANTAGES OF PROPOSED SYSTEM

* 1. SYSTEM REQUIREMENTS 18

3.3.1 HARDWARE REQUIREMENTS

3.3.2SOFTWARE REWUIREMENTS

* 1. LANGUAGE SPECIFICATION 19

## 4. SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE21

4.2 SEQUENCE DIAGRAM 22

4.3 USECASE DIAGRAM 24

4.4 CLASS DIAGRAM 25

## 5. SYSTEM IMPLEMENTATION

5.1 MODULES

5.1.1 STRUCTURAL TAGGING PROCESS 26

5.1.2 CHUNKING THE REVIEWS 29

5.1.3 BUILDING THESAURAS 34

5.1.4 SERVICE RECOMMENDER APPLICATION 36

## 6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION 39

6.2 FUTURE ENHANCEMENT 39

**APPENDIX 1 SAMPLE CODING** 40

**APPENDIX 2 SCREENSHOTS** 45

**REFERENCES** 48

**CHAPTER 1 INTRODUCTION**

**1.1 NATURAL LANGUAGE PROCESSING**

Computers are great at working with standardised and structured data like database tables and financial records. They are able to process that data much faster than we humans can. But us humans don’t communicate in “structured data” nor do we speak binary! We communicate using words, a form of unstructured data. When we program computers using something like C++, Java, or Python, we are essentially giving the computer a set of rules that it should operate by. With unstructured data, these rules are quite abstract and challenging to define concretely.

**APPLICATIONS OF NLP**

1. **SENTIMENT ANALYSIS**

Mostly used on the web and social media monitoring, NLP is a great tool to comprehend and analyse the responses to the business messages published on social media platforms. It helps to analyse the attitude and emotional state of the writer (person commenting/engaging with posts). This application is also known as opinion mining. It is implemented through a combination of Natural Language Processing and statistics by assigning values to the text (positive, negative or neutral) and in turn making efforts to identify the underlying mood of the context (happy, sad, angry, annoyed, etc.)



**Figure 1.1 Sentiment Analysis**

This application of NLP helps business organisations gain insights on consumers and do a competitive comparison and make necessary adjustments in business strategies, whenever required. Such data is also useful in designing a better customer experience and enhancing the product.

1. **CHATBOTS**

Chatbots are the solution for consumer frustration regarding customer care call assistance. They provide modern-day virtual assistance for simple problems of the customer and offload low-priority, high turnover tasks which require no skill. Intelligent Chatbots are going to offer personalised assistance to the customer in the near future. Many Industry analysts also predict that Chatbots are an emergent trend which will offer real-time solutions for simple customer service problems. They are unquestionably gaining a lot of trust and popularity from the consumer as well as engineers.

1. **CUSTOMER SERVICE**

Ensuring customer loyalty by keeping them content and happy is the supreme challenge and responsibility of every business organisation. NLP has aided in multiple functions of customer service and served as an excellent tool to gain insight into audience tastes, preferences and perceptions. Speech separation where the AI will identify each voice to the corresponding speaker and answer each of the callers separately. Excellent text to speech systems could even aid the blind. For example, a call recording of the customer can give insight into whether the customer is happy or sad, what are their needs and future requirements of the project .

* 1. **SYSTEM OVERVIEW**

In this work, we focus on modelling user-generated review and overall rating pairs, and aim to identify semantic aspects and aspect-level sentiments from review data as well as to predict overall sentiments of reviews. We propose a novel probabilistic supervised joint aspect and sentiment model (SJASM) to deal with the problems in one go under a unified framework. SJASM represents each review document in the form of opinion pairs, and can simultaneously model aspect terms and corresponding opinion words of the review for hidden aspect and sentiment detection. It also leverages sentimental overall ratings, which often come with online reviews, as supervision data, and can infer the semantic aspects and aspect-level sentiments that are not only meaningful but also predictive of overall sentiments of reviews. Moreover, we also develop efficient inference method for parameter estimation of SJASM based on collapsed Gibbs sampling. We evaluate SJASM extensively on real-world review data, and experimental results demonstrate that the proposed model outperforms seven well-established baseline methods for sentiment analysis tasks.

**1.3 SCOPE OF THE PROJECT**

## OBJECTIVE

The main aim is to build a new framework to identify semantic aspects and aspect level sentiments from users review data. The ability to correctly identify the sentiment expressed in user-reviews about a particular product is an important task for several reasons. First, if there is a negative sentiment associated with a particular product, the manufacturer can take immediate actions to address the issue. Failing to detect a negative sentiment associated with a product might result in decreased sales. From the users’ point-of-view, in online stores where one cannot physically touch and evaluate a product as in a real-world store, the user opinions are the only available subjective descriptors of the product. By automatically classifying the user reviews according to the sentiment expressed in them, we can assist the potential buyers of a product to easily understand the overall opinion about that product. Considering the numerous applications of sentiment classification such as opinion mining, opinion summarization, contextual advertising, and market analysis, it is not surprising that sentiment classification has received continuous attention. Sentiment classification can be considered as an instance of text classification where a given review must be classified into a pre-defined set of sentiment classes. In binary sentiment classification, a review must be classified into two classes depending on whether it expresses a positive or a negative sentiment towards an entity.

**CHAPTER 2**

**LITERATURE SURVEY**

**1)** **A Parser Generator for AI Languages**

**Author: James R. Kipps**

**Year: 2018**

One side effect of AI research has been the development of programming systems/languages as vehicles for experimentation and demonstration of concepts. These systems can be nontrivial to build. While parser generators can ease the task of language development, commonly available generators, such as YACC, use SLR, LR(l), and LL(1) parsing technologies that severely limit “human-friendly” syntaxes; they also do not interface to LISP. The RAND Compiler Kit (RACK) has been a parser generator for AI applications. RACK parsers have been unique in their ability to recognize non-LR(k) languages. RACK has been implemented in C and have been upwardly compatible with YACC. RACK generates parsers that interface with C.

**2) Semi-structured Chinese document analysis**

**Author: Zhang Chuang, Wu Ming, Li Chun Guang, Xiao bo, Lin Zhiqing**

**Year: 2017**

Semi-structured Chinese document analysis has been the most difficult task for complex structure and Chinese semantics. According to the generic characteristics of the semi-structured document and the specific characteristics of the resume document, the system researched on resume document block analysis based on pattern matching, multi-level information identification and feedback control algorithms has also prompted. Based on the research, resume parser system has implemented for China HR, which has the biggest recruitment Website. It can read, analysis, retrieval and store the information automatically. According to all kinds of experiments results, the accuracy and efficiency of this system can generally satisfy the practical requirements. As the research on the processing of the semi-structured document, it will not only be as a directive of the further research on the resume analysis, but also be as the reference to other form of the semi-structured document.

**3) Research and Implementation of Intelligent Chinese resume Parsing**

**Author: ZhiXiang Jiang, Chuang Zhang, Bo Xiao, ZhiQing Lin**

**Year: 2017**

Chinese Resume Parsing has been a method that extract information from kinds of online resumes by the method of statistical and rule-based. So, the companies can archive and manage the resumes automatically. Extracting personal information, such as name, gender, education, education background, work experience etc., from resumes and storing the information in the database, which requires a higher rate of precision and recall. And because of individual information (or personal basic information, such as name, contact information) offers the way that recruitment contact with the job-seekers, it needs a higher accuracy rate and recall. It can read, analysis, retrieval and store the information automatically. It describes an analytics-based propensity scoring model for re-skilling by combining historical employee job-role/skill records, relationships between different job-roles/skills, employee resumes, and job postings and much more parsing techniques are implemented efficiently.

# 4) Propensity modeling for employee Re-skilling

# Author: Moninder Singh, Karthikeyan Natesan Ramamurthy

**Year: 2017**

Due to the rapidly changing, dynamic nature of today's economic landscape, organizations are often engaged in a continuous exercise of matching their workforce with the changing needs of the marketplace. The Re-skilling offers these enterprises the ability to effectively manage and retain talent, while also satisfying business requirements. It describes an analytics-based propensity scoring model for re-skilling by combining historical employee job-role/skill records, relationships between different job-roles/skills, employee resumes, and job postings. This has been used to determine the source features that are the closest to a required target skill and hence identify employees that can be easily trained for the target skill.

**5)A Combined Semantic-Syntactic Sentiment Analysis for Students Assessment**

**Author: Vincenzo Cannella, and Roberto Pirrone**

**Year: 2016**

TutorJ is an Intelligent Tutoring System able to fulfil the requests of a student with a learning path inside didactical materials. To this aim, it must assess the level of training of the learner. In the first version of TutorJ this goal was reached through a conversational agent whose linguistic interaction enriched by a LSA-based text analysis. This approach suffers from the limitations of LSA as a bag-of-words approach. Next, morphosyntactic comparison of sentences' structures was implemented. Finally, it presents a new version of the assessment procedure involving both semantic, and morphosyntactic analysis user's sentences.

# 6) Automatic extraction of usable information from unstructured resumes to aid search

**Author:** [**Sunil Kumar Kopparapu**](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Sunil%20Kumar%20Kopparapu.QT.&newsearch=true)

**Year: 2016**

Automatic extraction of usable information from unstructured resume describes a system for automated resume information extraction to support rapid resume search and management. The system has capable extracting several important informative fields from a free format resume using a set of natural language processing (NLP) techniques. It has been described as a working system, for automatic resume management. The extension has six major extracting fields of information as defined by HR-XML. Experimental results carried out on large scale.

**7) Resume Parsing and Processing Using Hadoop**

**Author: Sourav Madhesiya**

**Year: 2016**

Big information would possibly be a gather of structured, semi structured and unstructured data that contain the massive amount of data, may be a private   
and academic data of person, that would help to screen out the candidates usually followed by an interview. This deals with the parsing application developed for the resumes received through emails in various formats like Document, text etc. The Resume computer program automatically utterly entirely different data on the various fields and parameters like name, mobile number, skills etc. and large volume of resumes have no drawback for this technique and each one work has completed automatically with none personal or human involvement.

**8) A Survey on Unstructured Text Analytics Approaches for Evaluation of Resumes**

**Author: Vinaka Ramesh Kudatakar**

**Year: 2015**

With the growing use of more and more data on networks, big data has become the new trend for productivity, innovation and competition across companies and industries. The proliferation of textual data in businesses has been overwhelming. Unstructured or semi-structured textual data is being constantly generated via web logs, emails, documents on the web, blogs, and so on. While the amount of textual data is increasing very rapidly, the ability to summarize, analyse which make sense of such data for making good business decisions.

# 9) Parsing natural language text of use case description

**Author:** [**Nakul Sharma**](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Nakul%20Sharma.QT.&newsearch=true)**,** [**Prasanth Yalla**](https://ieeexplore.ieee.org/search/searchresult.jsp?searchWithin=%22Authors%22:.QT.Prasanth%20Yalla.QT.&newsearch=true)

**Year: 2014**

Natural Language Text has always been subject to varied level of understandings and interpretations. It is always being a challenge to understand and develop interpretations from such a text. SDLC has artefacts which aid in completion of a successful software project. As many of these artefacts have textual data, they can be analysed for getting useful information both in regard to the content as well as the quality. Software Engineering process has textual description which can be analysed for generating useful information. Parsing natural language text of use case description an attempt has been made to parse the natural language text available in form of use case description. The result of parsing text has also presented herein.

# 10)A Study of Parsing Process on Natural Language Processing

**Author: Elisa Margareth Sibarani**

**Year: 2014**

Research on Natural Language Processing (NLP) in Indonesia has still limited and the results of available research that can be used for further research are also limited. In a series of natural language processing, the initial step is parsing the sentence in a particular language based on the grammar in order to help understanding the meaning of a sentence. This research aims to produce a simulation of Indonesian parser by adapting the process which was conducted by using Collins Algorithm. The three main stages are: 1) pre-processing to generate corpus and events files, 2) lexical analysis to convert the corpus into tokens, and 3) syntax analysis to build parse tree that requires file events to calculate the probability of the grammar by count the occurrence frequency on file events to determine the best sentence trees. An evaluation was performed to the parser using 30 simple sentences and the outcomes were able to generate a corpus file, file events, parse-tree and probability calculations. Nevertheless, some sentences could not be parsed completely true because of the limitations of the Tree bank file in Indonesian. Some future works are to develop complete and valid Tree bank and Lexicon files.

**CHAPTER 3**

**SYSTEM ANALYSIS**

## 3.1 EXISTING SYSTEM

In an existing probabilistic joint topic-sentiment (or sentiment-topic) models are unsupervised or weakly/partially supervised, meaning that they primarily model user-generated text content, and have not considered overall ratings or labels of the text documents in their frameworks. The previous model usually treats the overall sentiment analysis and aspect-based sentiment analysis in isolation, and then introduces a variety of methods to analyze either overall sentiments or aspect-level sentiments, but not both. Moreover, there is no group chunking process and a pre-defined word set to load. There can never be full scale understanding of positive and negative words until the verbal dictionary is added.

## 3.1.1 DISADVANTAGES OF EXISTING SYSTEM

* Unsupervised/partially supervised models
* Overall ratings were not considered
* Methods to analyse sentiments were used in isolation

### 3.2 PROPOSED SYSTEM

In our proposed system we develop supervised joint aspect and sentiment model to analyze overall and aspect-level sentiments for online user generated review data, which often come with labeled overall rating information. We demonstrate the sentiment classification for hotel domain. The implementation uses Natural Language Processing Techniques for extracting aspects and uses the Domain Thesaurus to classify the Aspects based on the Target Domains. Valance and Arousal will be calculated to calculate rating for the particular aspects in the user Review. A user-based CF algorithm is adopted to generate appropriate recommendations. It aims at calculating a personalized rating of each candidate service for a user, and then presenting a personalized service recommendation list and recommending the most appropriate services to him/her.

## 3.2.1 ADVANTAGES OF PROPOSED SYSTEM

* Supervised Models
* Usage of overall labelled information
* Usage of NLP techniques
* More Accurate results

**3.3 SYSTEM REQUIREMENTS**

## 3.3.1 HARDWARE REQUIREMENTS

* System: INTEL I3 and Above
* Hard Disk: 250 GB and Above
* Ram: 4GBand Above
* Operating system: 64-bit

## 3.3.2 SOFTWARE REQUIREMENTS

* Windows XP and Above
* JDK 1.7
* Tomcat 6.0
* MySQL 5.0
* Word Net

**3.4 LANGUAGE SPECIFICATION**

## 3.4.1 JAVA

Java technology

Java technology is both a programming language and a platform. The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

▪ Simple

▪ Architecture neutral

▪ Object oriented

▪ Portable

▪ Distributed

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed.   
  
The following figure illustrates how this works.

Every Java interpreter, whether it’s a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

The Java Platform

A platform is the hardware or software environment in which a program runs.

We’ve already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it’s a software-only platform that runs on top of other hardware based platforms.

The Java platform has two components:

* The Java Virtual Machine (Java VM)
* The Java Application Programming Interface (Java API)

Introduced to the Java VM it’s the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 SYSTEM ARCHITECTURE**

Admin

File System

CSV Files

Batch Processing

Tagged And Chunked Files and Collections

WE

B

S

E

RV

I

C

E

Post

Review

Top Rated

Hotels

WE

B

S

E

RV

I

C

E

NLP Process

Plan a Travel

Similarity

Computation

Aspect Ratings

Analysis

User

Recommendation

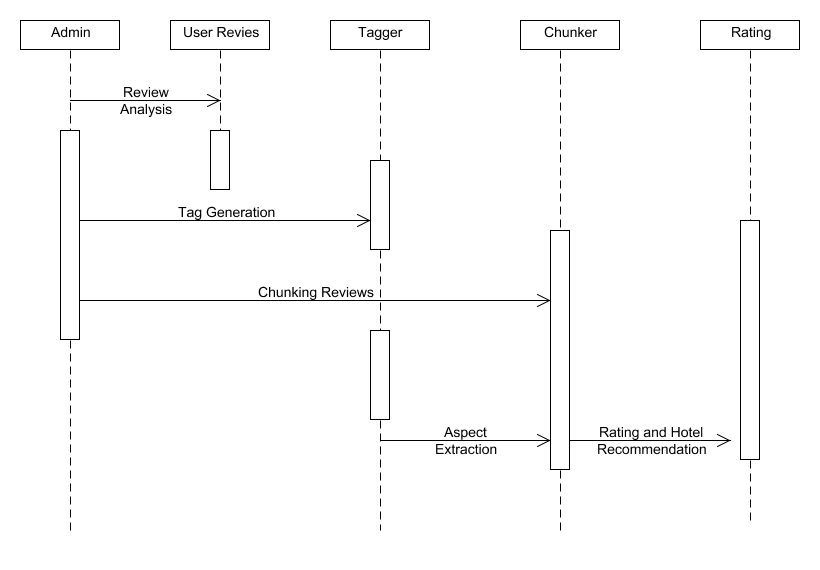
**Figure 4.1 System Architecture**

A system architecture is the conceptual model that defines the structure, behaviour and more views of the system. An architecture description is a formal description and representation of system, organized in a way that supports reasoning and behaviours of the system.

In figure 4.1 the user reviews and ratings from the user side will be given as input in the form of datasets to the admin side. Normally the users after using a Hotel he will rate it and give reviews about the hotel online , the top reviews will be collected and categorized using the NLP process and will be send to the admin side in the form of datasets. The admin will import the datasets in the csv file format and perform the batch processing. The input will be tagged and chunked and then the recommendations will be generated and send to the user side.

Now if a user plans a travel based on his preferences similarity computation will done to fetch the dataset matching to his preferences. After that Aspect ratings will be done to give the recommendations about the hotels to the users.

**4.2 SEQUENCE DIAGRAM**



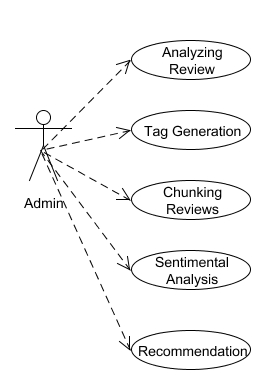
**Figure 4.2 Sequence diagram**

In figure 4.2 the Sequence diagrams are used to capture the order of messages flowing from one object to another. There are two class roles namely server and client, that describe the way an object will behave in context. An activation is represented by a thin rectangle on a class which represents the period during which an element is performing an operation.

The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively. A call message defines a particular communication between lifelines of an interaction, which represents an invocation of operation of target lifeline. A return message defines a particular communication between lifelines of an interaction, which represents the pass of information back to the caller of a corresponded former message.

Initially the admin starts the request analysis and gets the user reviews for input. Then the admin calls the tagger process to tag the user reviews. Now the admin will call Chunking process and the it will get the input from the tagger process using aspect extraction. Finally the Rating and Hotel Recommendation will be send to the Rating process.

**4.3 USE CASE DIAGRAM**



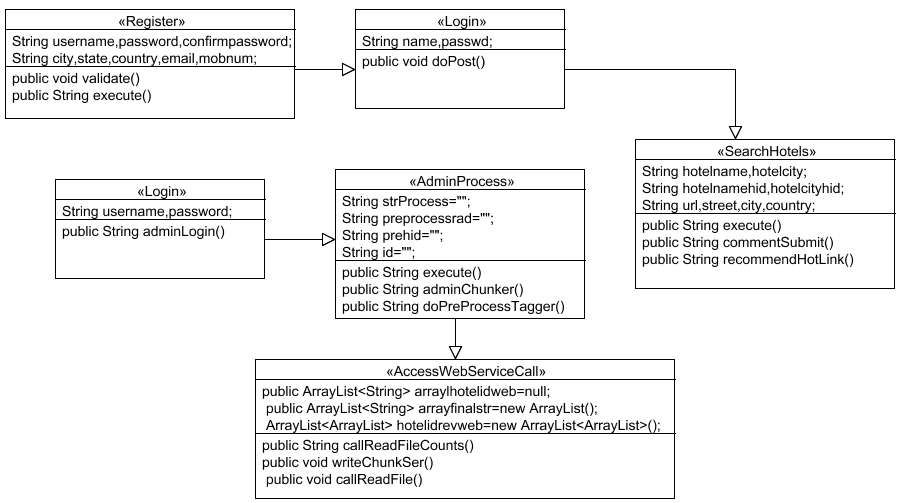
**Figure 4.3 Use case diagram**

A use case diagram is a graphic depiction of the interaction among elements of system. And as with all Use Case diagrams, Actors must be represented as stick people. Each class of interaction is represented as a named eclipse. The named eclipse represents the description of system behaviour as it responds to a request that originates from outside the system.

In figure 4.3 the actor of the system is admin. Client represents the admin. Initially the admin sends a request to start the analysing of review. Then the input is uploaded for tag generation and then the chunking of reviews based on the output from the previous process. Now the admin will call the sentimental analysis process to produce the recommendations and displayed when the admin sends signal to Recommendation process. This works on mapping the modules of Analysing the review, Tag generation, Chunking Review, Sentimental Analysis and Recommendations. Each and every module has its own process in the data mining.

And as with all Use Case diagrams, Actors must be represented as stick people. Each class of interaction is represented as a named eclipse. The named eclipse represents the description of system behaviour as it responds to a request that originates from outside the system. They work sequentially to give output in such a way that the command prompt doesn’t throw any error or warnings.

**4.4 CLASS DIAGRAM**



**Figure 4.4 Class diagram**

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. They are widely used in modelling of object-oriented systems which can be mapped directly with object-oriented languages.   
It shows a collection of classes, interfaces, associations, collaborations, and constraints.

In figure 4.4 there are two set of classes. In one set there are three classes Register, Login and Search Hotels which is connected by unidirectional association. In the register class there are several attributes like username, password, confirm password, city, state, country, mobnum, email to perform operations like validate and execute. In the Login class there are two attributes username and password to perform dopost operation. In the search hotels class there are attributes like hotelname, hotelid, hotelcity, hotelcityid, hotelstreet and hotelcountry to perform operations like execute, comment submit and recommend hotel link.

In another set there are three classes Login, Admin Process and Access Web Service which is connected by unidirectional association. In the Login class there are two attributes username and password to perform admin login operation. In the admin process class there are attributes like strprocess, pre-process, id etc to perform some operations like execute, admin Chunker and do pre process tagger. In the Access Web Service class there are some attributes like array hotel id, array finalstr, and hotelIdreWeb to perform some operations like call read file counts, write Chunk and call read file.

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**5.1 MODULES**

The modules are

1. Structural Tagging Process.
2. Chunking the Reviews.
3. Building Thesaurus.
4. Service Recommender Application

## 5.1.1 STRUCTURAL TAGGING PROCESS

In this module data are retrieved from open source datasets that are publicly available from major Travel Recommendation Applications. The CSV(Comma separated values) files were read and manipulated using Java API that itself developed by us which is developer friendly ,light weighted and easily modifiable.

## 5.1.2 CHUNKING THE REVIEWS

Chucker Process is done on each and every review of all and the products. The Clunker Process will take POS tagged output as input for Grouping the Words based on meaning of the Review. Chunker Process is done so that we can easily extract the sentiment embeddings associated with the Aspects of the particular review. The meaningful words that should be read continuously for proper understanding of the review are marked with square bracket. Now the Aspects in each review are extracted from the POS Tagger result.

The Noun and Phrasal Verbs are the key Attributes in any sentence. So those things were extracted from the tagged reviews and marked as Aspects of the particular review by a user. Now mappings are done to properly annotate the user review and associated Aspects with the Chunks in it. Online reviews are an important source of product information that influences consumers’ product attitude and purchase behaviour. This research proposes review chunking, that is, grouping reviews by valence, as a newly recognized factor that influences the persuasive effect of online reviews. In particular, this study examines the differential effects of review chunking on product attitude for consumers with high versus low motivation to think. The results from two experiments show that for consumers with low motivation to think, review chunking has a negative effect on product attitude. For consumers with high motivation to think, the effect of review chunking depends on whether they read positive or negative reviews first. These findings extend our understanding of the factors that influence the persuasiveness of online reviews and provide practical implications to online retailers on how to influence consumers’ product attitude by changing the default criteria of the sorting and filtering aids on their online review systems.

## 5.1.3 BUILDING THESAURAS

A Domain Thesaurus is built depending on the Keyword Candidate List and Candidate Services List. Keyword Candidate List is the and Candidate Services List are interdependent on the Target domains and it can be prepared before porting the classifier to Target domain. Expert Knowledge should be given for preparing the domain Thesaurus. The Domain Thesaurus can be Updated Regularly to get accurate Results of the Recommendation System. Now the Aspects extracted are subjected to domain groping based on the target domain.

## 5.1.4 SERVICE RECOMMENDER APPLICATION

The Chunked Reviews of the Similar User List is retrieved and the Keywords corresponding to the User is analyzed for its Valence and Arousal. Valence Means Weather the Keywords Means a positive or Negative thing and Arousal answers, how much it is? Ratings are given for each Domain based on the Valence and Arousal for each User of each hotel. The Overall Hotel Rating is now manipulated by taking average values of each rating of several users of a particular hotel. Now ranking is done for all hotels based on Ratings and will be sorted based on Bubble Sort Algorithm to have the Most appropriate personalized Recommendation for the User. The Results will be analyzed with Graphical Views so as to understand easier. The domain thesaurus plays an important role in information retrieval, natural language processing, question answering system etc. Due to the complexity of the natural language, the NLP based thesaurus constructing methods are difficult to achieve a desired result. In recent years, Wiki has been widely used as a knowledge base. Based on the characteristics anchor description and topic locality of hyperlinks, this paper proposes a hyperlink structure graph clustering based domain thesaurus construction method. The method first constructs a domain-specific hyperlink structure graph using Wiki, and then uses LSI algorithm to calculate the weight of each hyperlink. Then our method uses CPMw algorithm to cluster the weighted undirected hyperlink structure graph. After this step, domain thesaurus can be achieved. Experiments show that our method can get better results.

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

Hence, we proposed and developed a framework to identify semantic aspects and aspect level sentiments from users review data. The output will be more presentable with a user interface thus the web development concepts are implemented to portray the results in a localhost page. The results in the page works on the flow from registration of user. Then the respective user can sign in the web page. The pre requisites are the pre-processing and chunking process to be done for the data set. The data set comprises of several cities from the world and its hotel reviews. These reviews are carefully sorted into positive and negative words using Wordnet software inbuilt. After the separation of the words, the requirements are obtained in the webpage. Collecting the requirements, the hotel reviews are filtered accordingly and displayed. In addition to this, the user can add his review as well and it will be added in the data set sequentially.

**6.2 FUTURE ENHANCEMENT**

The Natural Language Processing is implemented to analyze the reviews of the previous user. The NLP Process Comprises Tokenizing a Sentence or a word, POS (Parts of Speech) Tagging, Extraction of Nouns and Verbs, Synonym Retrieval and Spell Check of Extracted Keywords using WordNet Dictionary. Valence and Arousal will be implemented for calculating Ratings of Aspects of a Hotel.

**APPENDIX 1**

**SAMPLE CODING**

**CHUNKER:**

public List chunkSentence(List words, List tags, List pos)

{

words.add("ZZZ");

pos.add("ZZZ");

tags.add("Z");

Iterator it = rules.iterator();

while (it.hasNext())

{

List newTags = new ArrayList();

Rule r = (Rule)it.next();

for (int i = 0 ; i < words.size() ; ++i)

{

if (r.match(i,words,tags,pos))

{

newTags.add(r.getNewTag());

}

else

{

newTags.add(tags.get(i));

}

}

tags = newTags;

}

words.remove(words.size()-1);

pos.remove(pos.size()-1);

tags.remove(tags.size()-1);

**RULE:**

public boolean match(int currentToken, List words, List tags, List pos)

{

if (!withinSentence(words.size(), currentToken)) return false;

boolean matched = true;

for (int i = 0 ; i < types.size() ; ++i)

{

String type = (String)types.get(i);

List ofs = (List)offsets.get(i);

String value = (String)values.get(i);

List working = null;

if (type.equals("T"))

{

working = tags;

}

else if (type.equals("W"))

{

working = words;

}

else if (type.equals("P"))

{

working = pos;

}

for (int j = 1 ; j < ofs.size() ; ++j)

{

offset = ((Integer)ofs.get(j)).intValue();

matchOffset = matchOffset || ((String)working.get(currentToken+offset)).equals(value);

}

matched = matched && matchOffset;

if (!matched) i = types.size();

}

return matched; }

**STEMMER:**

**public Vector Stem ( Vector words ){**

## if ( !IsInitialized )

## return words;

## for ( int i = 0; i < words.size(); i++ )

## {

## words.set( i, Stem( (String)words.get( i ) ) );

## }

## return words;

## }

## public Vector processStem(String wor)

## {

## Vector v=new Vector();

## v.add(wor);

## Vector wrds=Stem(v);

## System.out.println("out wrds + " +wrds );

## return wrds;

## }

## }

**WORDNET CONNECTION:**

public class WordnetConnection

{

WordNetDatabase database;

public WordnetConnection()

{

try

{

Properties prop=new Properties();

FileInputStream fis=new FileInputStream(new File("webapps/CrossDomain/diction.properties"));

prop.load(fis);

String path=prop.getProperty("dictpath");

System.setProperty("wordnet.database.dir",path.trim());

database = WordNetDatabase.getFileInstance();

}

catch(Exception e)

{

e.printStackTrace();

}

}

**DOMAINTHESPROD:**

public class DomainThesProd

{

public LinkedHashMap<String,Vector> domainlinkmap=new LinkedHashMap<String,Vector>();

public LinkedHashMap<String,Integer> domainindexmap=new LinkedHashMap<String,Integer>();

public DomainThesProd()

{

Vector<String> cleanliness = new Vector<String>();

cleanliness.add("dirty");

cleanliness.add("grubby");

cleanliness.add("clean");

cleanliness.add("neat");

cleanliness.add("pure");

cleanliness.add("tidy");

cleanliness.add("shine");

cleanliness.add("dirtless");

cleanliness.add("hygienic");

cleanliness.add("fresh");

cleanliness.add("elegant");

Vector<String> environment = new Vector<String>();

environment.add("modern");

environment.add("environment");

environment.add("comfortable");

environment.add("classical");

environment.add("climate");

environment.add("surroundings");

environment.add("ambiance");

environment.add("situation");

environment.add("background");

environment.add("scenery");

environment.add("neighborhood");

environment.add("location");

environment.add("bright");

}

domainlinkmap.put("clean",cleanliness);

domainlinkmap.put("environment",environment);

domainlinkmap.put("familyfriends",familyfriends);

domainlinkmap.put("fitness",fitness);

domainlinkmap.put("food",food);

domainlinkmap.put("room",room);

domainlinkmap.put("service",service);

domainlinkmap.put("shopping",shopping);

domainlinkmap.put("transportation",transportation);

domainlinkmap.put("value",value);

domainlinkmap.put("entertainment",entertainment);

domainindexmap.put("clean",0);

domainindexmap.put("environment",1);

domainindexmap.put("familyfriends",2);

domainindexmap.put("fitness",3);

domainindexmap.put("food",4);

domainindexmap.put("room",5);

domainindexmap.put("service",6);

domainindexmap.put("shopping",7);

domainindexmap.put("transportation",8);

domainindexmap.put("value",9);

domainindexmap.put("entertainment",10);

**ADMIN PROCESS (Attributes):**

public class AdminProcess extends ActionSupport implements MapInter

{

String strProcess="";

String preprocessrad="";

String prehid="";

String id="";

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getPreprocessrad() {

return preprocessrad;

}

public void setPreprocessrad(String preprocessrad) {

this.preprocessrad = preprocessrad;

}

AccessWebServiceCall accesswebcall=new AccessWebServiceCall();

public String getPrehid() {

return prehid;

}

public void setPrehid(String prehid) {

this.prehid = prehid;

}

public String getStrProcess() {

return strProcess;

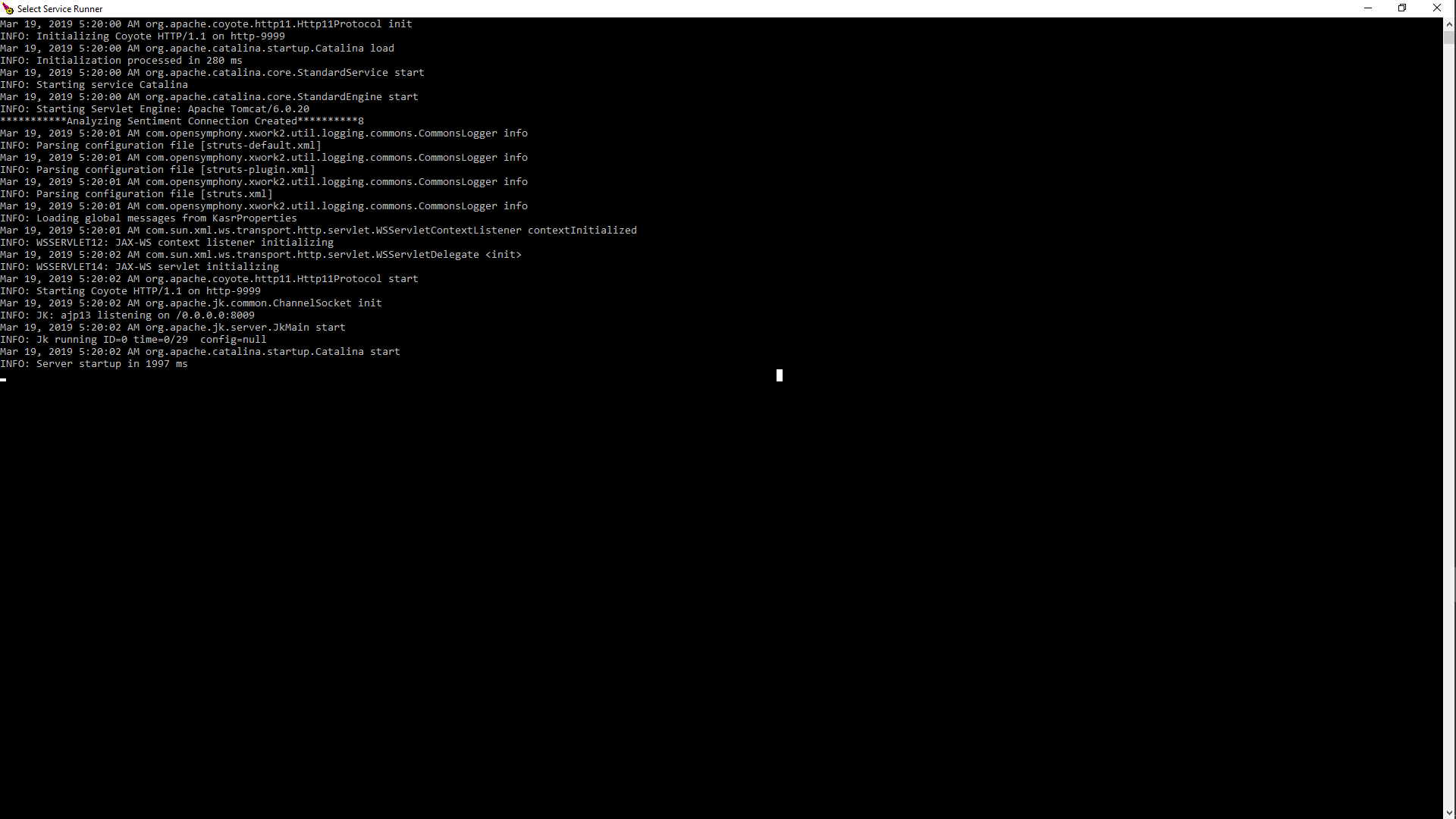
}

public void setStrProcess(String strProcess) {

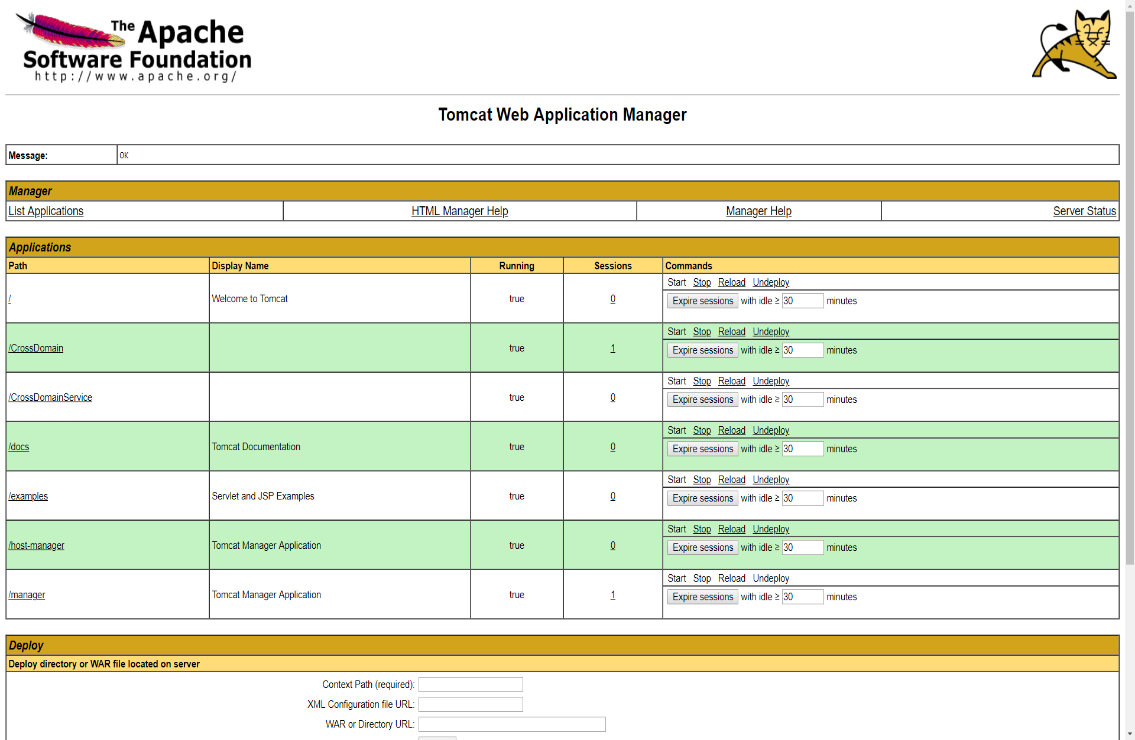
this.strProcess = strProcess;

}

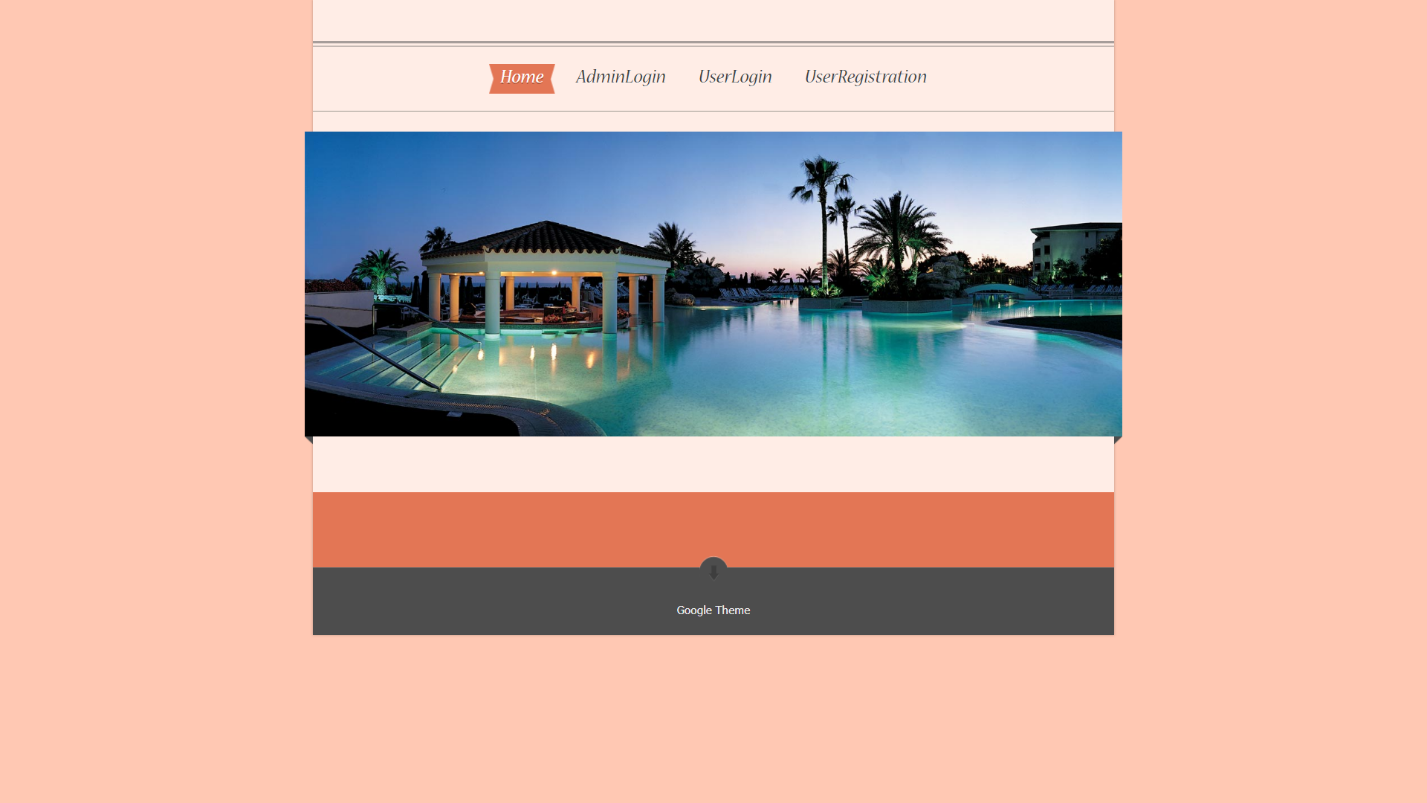
**APPENDIX 2 SCREENSHOTS**



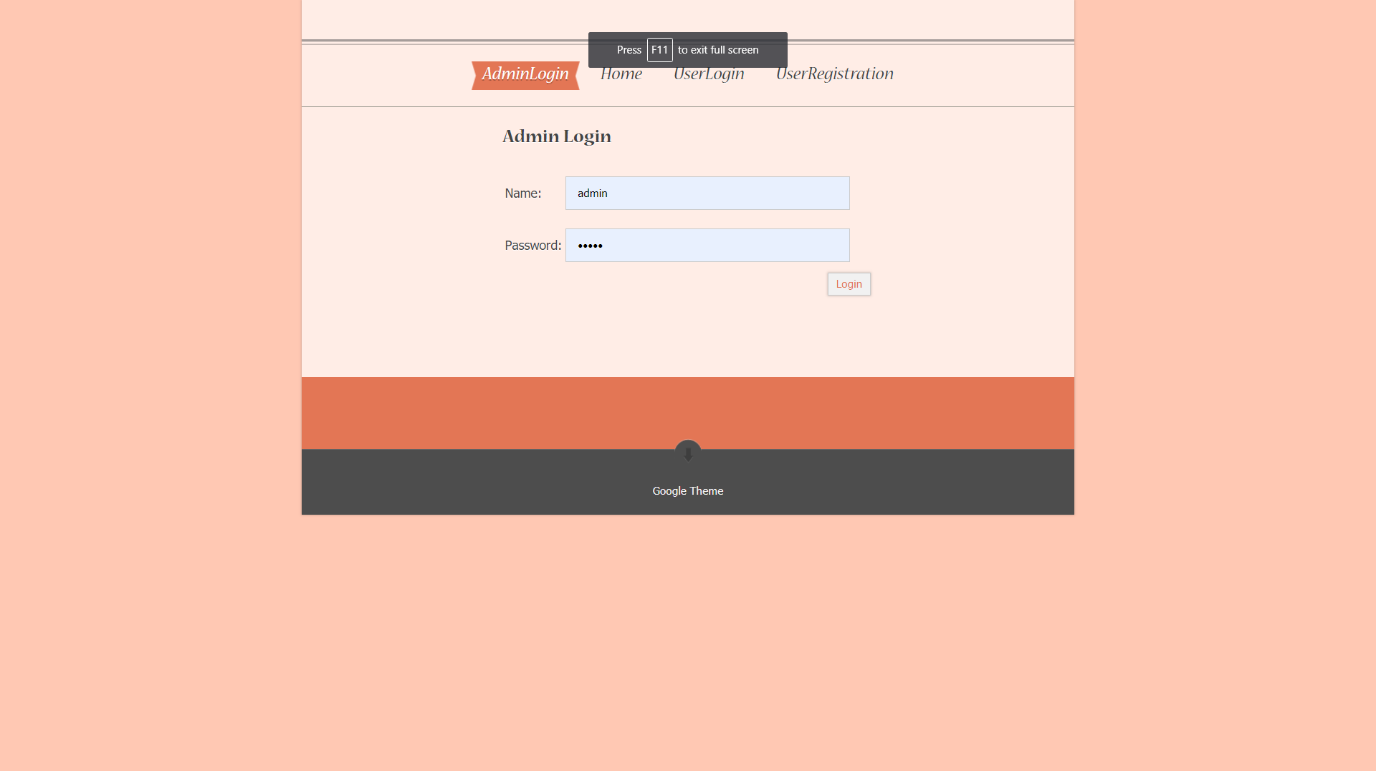
**STARTING THE SERVER**



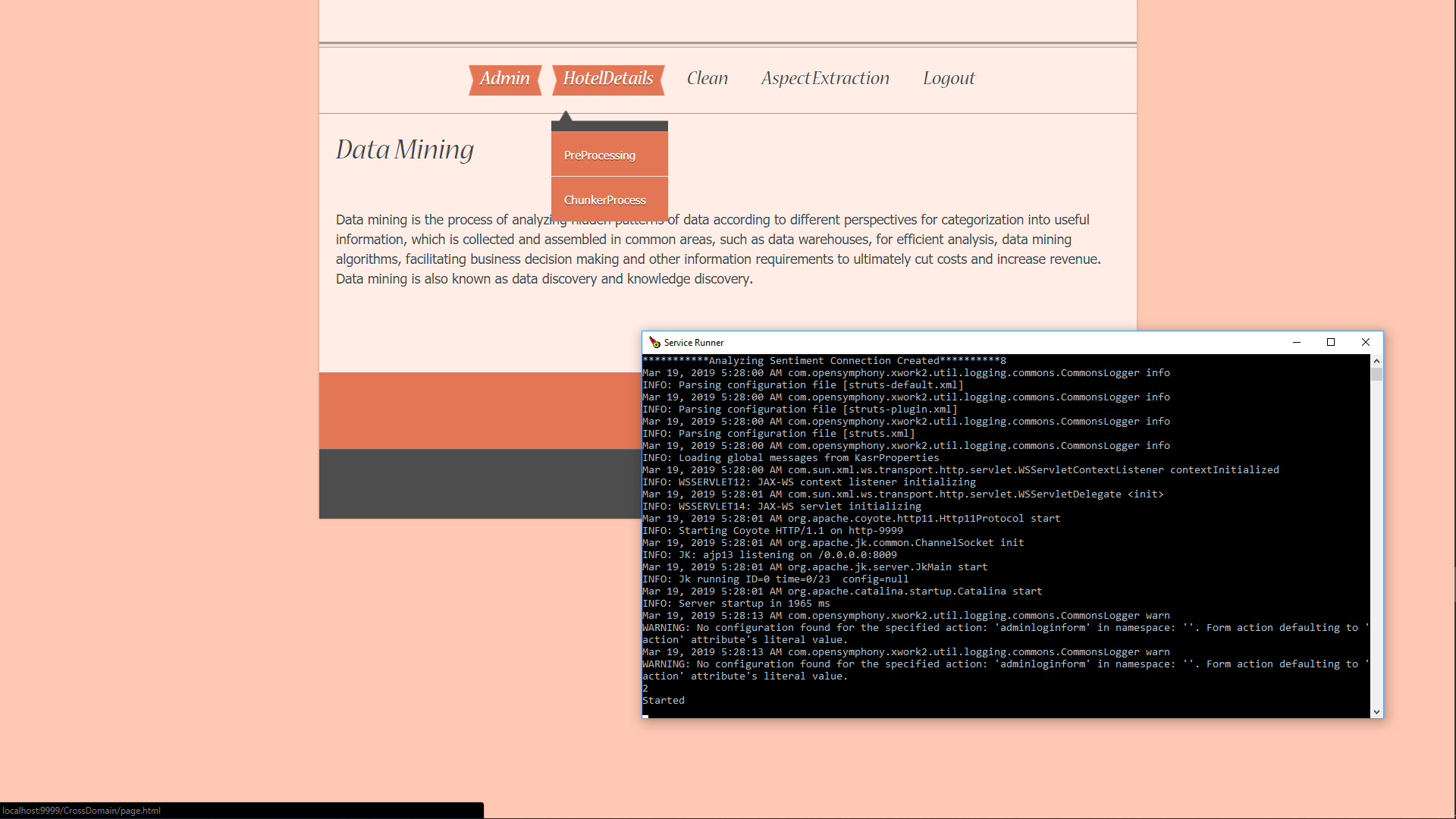
**TOMCAT WEB APP MANAGER**



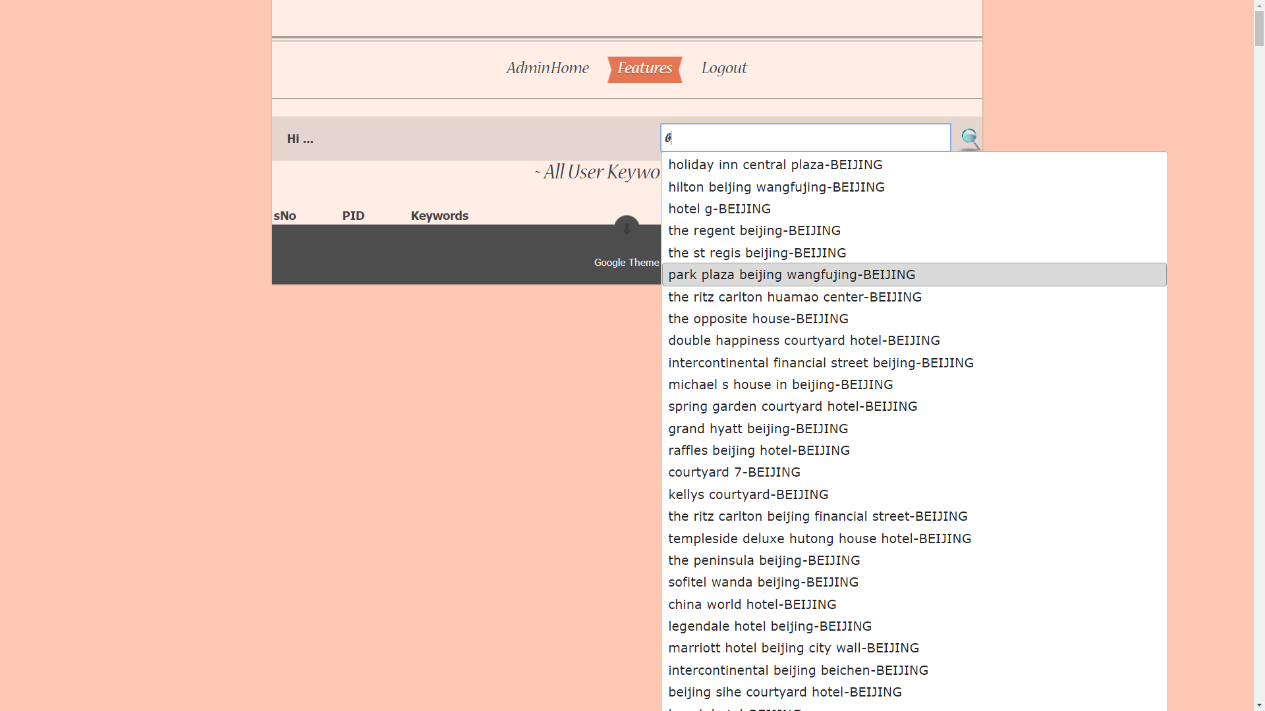
**USER INTERFACE**



**LOGIN PAGE**



**PREPROCESSING AND CHUNKING**



**ASPECT EXTRACTION**

**REFERENCES**

1. B. Liu, “Sentiment analysis and opinion mining,” Synthesis Lectures Human Language Technol., vol. 5, no. 1, pp. 1–167, May 2012.
2. B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: Sentiment classification using machine learning techniques,” in Proc. ACL-02 Conf. Empirical Methods Natural Language Process., 2002, pp. 79–86.
3. V. Ng, S. Dasgupta, and S. M. N. Arifin, “Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews,” in Proc. COLING/ACL Main Conf. Poster Sessions, 2006, pp. 611–618.
4. J. Zhao, K. Liu, and G. Wang, “Adding redundant features for CRFs-based sentence sentiment classification,” in Proc. Conf. Empirical Methods Natural Language Process., 2008, pp. 117–126.
5. P. Melville, W. Gryc, and R. D. Lawrence, “Sentiment analysis of blogs by combining lexical knowledge with text classification,” in Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2009, pp. 1275–1284.
6. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in Proc. 49th Annu. Meet. Assoc. Comput. Linguistics Human Language Technol., 2011, pp. 142–150.
7. R. Socher, J. Pennington, E. H. Huang, A. Y.Ng, and C. D.Manning, “Semi-supervised recursive autoencoders for predicting sentiment distributions,” in Proc. Conf. Empirical Methods Natural Language Process., 2011, pp. 151–161.
8. D. Tang, B. Qin, and T. Liu, “Learning semantic representations of users and products for document level sentiment classification,” in Proc. 53rd Annu. Meet. Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Language Process., Jul. 2015, pp. 1014–1023.
9. A. Abbasi, H. Chen, and A. Salem, “Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums,” ACM Trans. Inf. Syst., vol. 26, no. 3, pp. 12:1–12:34, Jun. 2008.
10. Agarwal, F. Biadsy, and K. R. Mckeown, “Contextual phraselevel polarity analysis using lexical affect scoring and syntactic Ngrams,” in Proc. 12th Conf. Eur. Chapter Assoc. Comput. Linguistics, 2009, pp. 24–32.