A PROJECT
ON

Scribe - Natural Language Understanding and Generation Engine

BY

Vandita Shah B-827
Sumer Shende B-834
Vineet Trivedi B-848
Rishabh Vig B-855

Under the guidance of

Internal Guide
Prof. Dnyaneshwar Dhangar

MCT
MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY
Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53

Department of Computer Engineering

University of Mumbai

Apr – 2016

MCT

MANJARA CHARITABLE TRUST

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

Juhu-Versova Link Road Versova, Andheri(W), Mumbai-53

CERTIFICATE

Department of Computer Engineering

This is to certify that

1. Vandita Shah B-826
2. Sumer Shende B-834
3. Vineet Trivedi B-848
4. Rishabh Vig B-855

Have satisfactory completed this project entitled

“Scribe - Natural Language Understanding and Generation Engine”

Towards the partial fulfillment of the

BACHELOR OF

ENGINEERING IN

(COMPUTER ENGINEERING)

as laid by University of Mumbai.

Guide	H.O.D.
Prof. Dnyaneshwar Dhangar	**Dr. S. B. Wankhade**

Principal
Dr.Udhav Bhosle

Internal Examiner	External Examiner

Project Report Approval for B. E.

This project report entitled “Scribe - Natural Language Understanding and Generation Engine” by Vandita Shah, Sumer Shende, Vineet Trivedi and Rishabh Vig is approved for the degree of ***Bachelor of Computer Engineering***.

External Examiners

1.----------------------------------

2.----------------------------------

Internal Examiner

1.----------------------------------

2.-------------------------------

HOD

Date:
Place

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken whenneeded.

Date:

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to Dr. U. V. Bhosle, Principal and Dr. S. B. Wankhade, H.O.D of Computer Department of RGIT for providing us an opportunity to do our project work on “Scribe - Natural Language Understanding and Generation Engine".

This project bears on imprint of many peoples. We sincerely thank our project guide Prof. Dnyaneshwar Dhangar for his/her guidance and encouragement in carrying out this synopsis work.

Finally, we would like to thank our colleagues and friends who helped us in completing the Project work successfully.

ABSTRACT

"SCRIBE" is an AI driven system which automatically generates output in a user defined format based on raw data that is fed to it. It is a tedious and progressively difficult task for a human to process large volumes of data and deliver meaningful output in a scenario where things are changing by the minute. Thus "SCRIBE" reorganizes raw data into furnished output as required by the user whether it be in the form of an essay, article, post or letter. "SCRIBE" collects raw data and stores it in a file system. It applies adequate business rules for data cleaning, standardization and mining.

The polished data is then formatted as per user requirement using narrative science. In essence "SCRIBE" approaches the highest epoch of artificial intelligence to automate creative human tasks like writing.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION AND MOTIVATION

1.1

FOREWORD

“SCRIBE” is an AI driven system which automatically generates output in a user defined format based on raw data that is fed to it. It is a tedious and progressively difficult task for a human to process large volumes of data and deliver meaningful output in a scenario where things are changing by the minute. Thus “SCRIBE” reorganizes raw data into furnished output as required by the user whether it be in the form of an essay, article, post or letter. “SCRIBE” collects raw data and stores it in a file system. It applies adequate business rules for data cleaning, standardization and mining.

The goal of the system is to design and build software that will analyze, understand, and generate languages that humans use naturally. For this we use the concept of Natural Language Processing (NLP), which is a field of computer science concerned with interactions between computers and human languages.

1.2 INTRODUCTION TO SCRIBE

The system takes raw data as input and converts it into sentences that may form a small article. But before accepting the raw data the system is taught to understand basic English language syntaxes and the structures of sentences by analyzing various text documents and find links between commonly used words. This is done so that the system knows the difference between various parts of speech. It helps the system break the sentences into elemental components which it saves in the database as links. These links exist between a word and its successive word. Ultimately each word will have a list of words that could follow it, which in turn comes in handy while forming sentences as the system will consider all possible successive words and select the best one based on various factors such as tense, tone etc. Once the system has analyzed the text documents it can efficiently segregate the data in a way that is useful for sentence formation. This entire process is carried out by following the concepts of natural language processing.

After understanding the proper grammar syntaxes and forming the links between words, we move on to the part where the user gives some raw data as input, this could be a set of random words all related to some specific topic. SCRIBE takes this input and finds the link for each word from the input in the database, it then checks the probability of the word and its successive word so as to select the appropriate word for the final sentence. This is done by using tokens and lexical analysis which are a sub topic in natural language generation. The final output would consist of a set of sentences made on the basis of the data entered by the user.

1.3 MOTIVATION

The main motivation behind the project was to eliminate human delay from the process of effective communication. During natural disasters chances of human delay are high and the need to communicate the disaster and its consequences is of paramount importance. Abnormal readings from instruments like seismometers, Saffir-Simpson scale and stream gauges indicating a natural disaster like earthquake, hurricane or flood respectively should be communicated to Scribe. Scribe can then write a piece informing the people of a natural disaster. Scribe can be one of the fastest modes of communicating the occurrence of a natural disaster, the area it has occurred in and its possible impact.

CHAPTER 2

PROBLEM STATEMENT

It is a repetitive and dynamically difficult errand for a human to process substantial volumes of raw data being perceived continuously from different domain, understand and interpret it while creating large amounts of text compositions which are syntactically correct and convey the logic preserving the original meaning.

Scribe is made up of two main modules – Learning modules based on natural language processing; which reads data from thousands of well drafted articles, understands the syntax and stores this information in a knowledge base. This knowledge is later used for NLG to obtain rich output compositions.

The second module of Scribe is the natural language generation module which consists of two main components: the first carries out Analysis and Interpretation, and the second is responsible for Information Delivery. The Analysis and Interpretation stage derives facts and insights from the raw input data and turns them into basic information chunks called messages. The Information Delivery stage works out how to best communicate the information in these messages as a coherent text.

This system revamps crude information into text articles as required by the user. It gathers raw data from sources like:-

1. RSS feeds of multiple websites
2. Social media plug roots
3. Blogging stations
4. News Tickers

For this, Scribe needs to be able to read text from different formats like XML, LateX along with grabbing the correct unformatted text from html documents.

The three key advantages of Scribe-produced text are scalability, tailor ability and consistency. It can produce personalized text at a rate that is simply unattainable using human authors, and the quality of NLG texts does not vary in the way that a human's might. Scribe is ideal for anyone who-

- has data that needs to be interpreted in order to make it actionable
- wants to obtain quick text composition from raw data
- needs to make fast decisions based on data
- has too much data for a human expert to understand and interpret
- requires consistency in their text output
- wants to make their data accessible to people from different domains.

For the implementation of this system, we require a large database that is used to store the entire word POS relations obtained from scanning diverse documents during the NLP phase. This data is stored in an unstructured relational form to make the data fetch more efficient.

In the NLG phase, it is crucial that the system understands the raw data. This can be done by referring its knowledge base to find relevant connections data items and reproduce them to formulate logical sentences. The NLG then should be able to produce grammatically correct sentences using language libraries and finally produce good quality reports.

Lastly, the user should be able to change the text into different formats. User should also be able to give a feedback to the system. This feedback will help the system to calculate its efficiency and check if it is giving satisfactory outputs. Feedback also needs to register in the knowledge base for future use by Scribe.

CHAPTER 3

REQUIREMENT ANALYSIS

After doing an extensive research on the requirements of a computer running an NLP process we came up with the following hardware requirements:

1. SAN storage network interface.
2. Solid State Hard drive.
3. Computer for user interface.

Apart from the hardware we required:

1. An NLP Library
2. An NLU Library
3. A schema free database to store semi structured data
4. A repository of cricket articles to serve as training data
5. Sources to extract information for the input article
6. Java JRE 6+
7. Eclipse
8. Windows/Linux Operating Environment

CHAPTER 4

PROJECT DESIGN

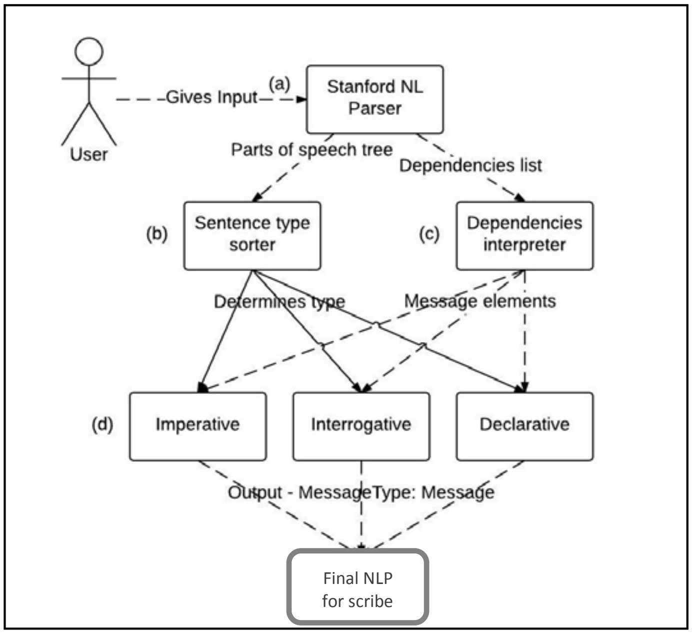


Diagram 4.1 Overall Design

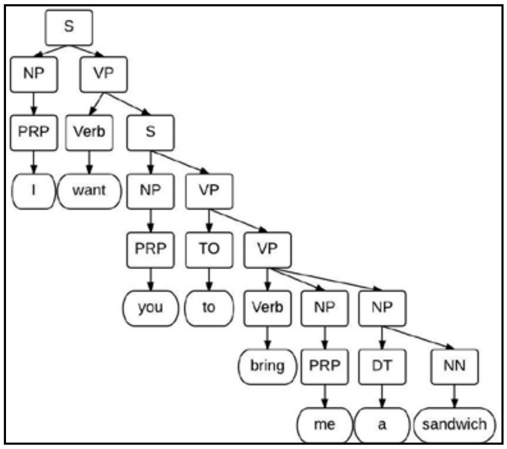
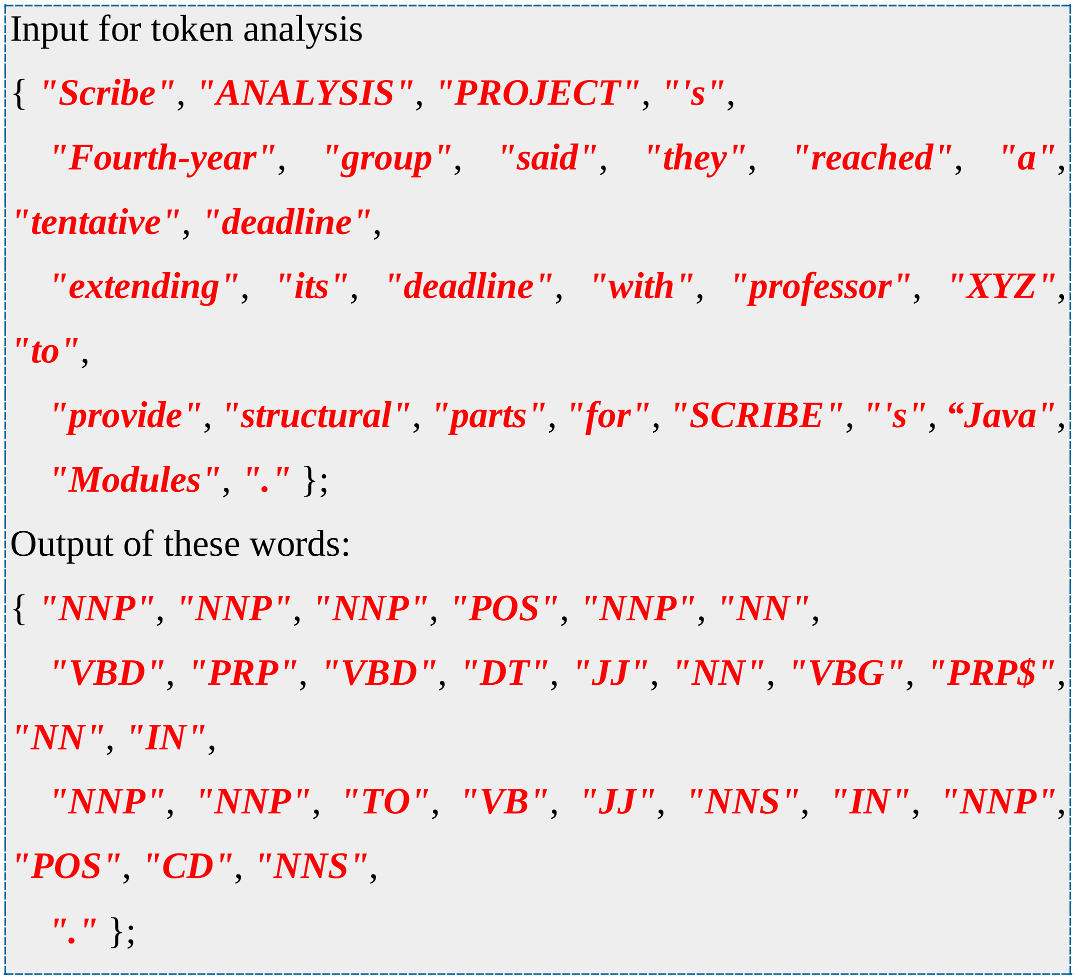


Diagram 4.2 Breakdown of sentences:



Input for token analysis

```
{ "Scribe", "ANALYSIS", "PROJECT", "'s",
   "Fourth-year", "group", "said", "they", "reached", "a",
"tentative", "deadline",
   "extending", "its", "deadline", "with", "professor", "XYZ",
"to",
   "provide", "structural", "parts", "for", "SCRIBE", "'s", “Java",
   "Modules", "." };
```

Output of these words:

```
{ "NNP", "NNP", "NNP", "POS", "NNP", "NN",
   "VBD", "PRP", "VBD", "DT", "JJ", "NN", "VBG", "PRP$",
"NN", "IN",
   "NNP", "NNP", "TO", "VB", "JJ", "NNS", "IN", "NNP",
"POS", "CD", "NNS",
   "." };
```

Formation of sentences to create complete article:

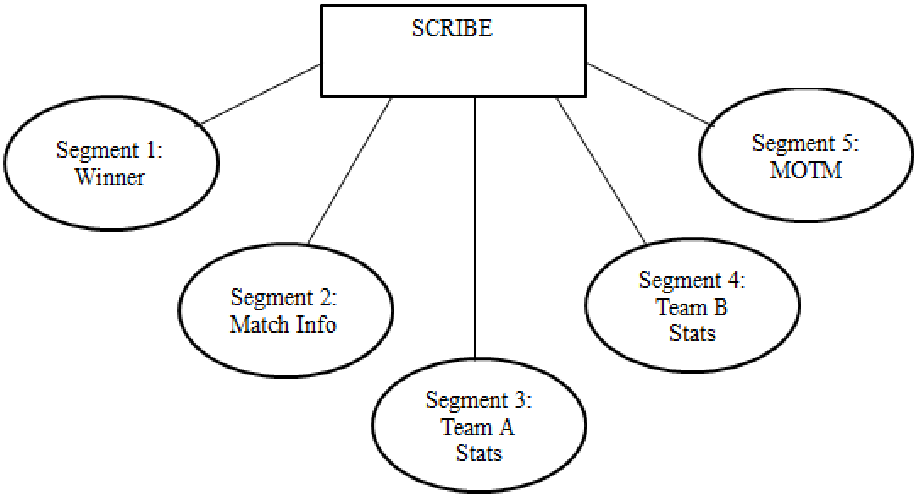


Diagram 4.3 Senctence formation

CHAPTER 5

IMPLEMENTATION DETAILS

The project is divided into the following phases:

1. Learning phase
2. Knowledge Implementation phase

Phase 1:

In this phase we teach the system about the English language, the structure of sentences and characteristics of each word. The aim is that the system should develop a correct grammatical sense. We use open source libraries like open nlg and Stanford nlg for this purpose. These libraries help establish links between words and categorise the words into nouns, pronouns, verbs, adverbs etc.

Once the system is equipped to understand the language we feed data to it. Numerous texts related to cricket are fed into the system. The libraries then breakdown these articles into sentences and provide the structure of each sentence. Thus the system learns how sentences are formed, which sentence structure is used most often and under what circumstances. This is the growth part where the system reads the documents and garners experience from them.

Phase 2:

Next phase is the Knowledge Implementation phase. Here the system uses all the knowledge it has acquired through the learning and growth phase to create the article.

An article needs to provide the following primary information winning team, match details, runs scored by each team, man of the match or any other outstanding performance by a player. So the system will try to convey this information. It will use a sentence to convey each point. The structure of this sentence is decided by experience. If a team has scored 300 runs in 50 overs and 7 wickets then the most frequently used structure type matching this scenario is used. The most frequently used structure type is obtained from articles fed to the system during the learning and growing phase.

The system creates a sentence to convey each of the above points and when we put all the sentences together we have the article in its entirety.

CHAPTER 6

TECHNOLOGIES USED

Hardware Requirements

- SAN storage network interface.

  This storage is used to store data from the automatic identification, capture, and summary from web pages, knowledge bases, emails, chats, social media, transcripts, conversations, and other unstructured formats.
- Solid State Hard drive.

  Databases can benefit from solid state storage. This can reduce the access speeds and querying delay by almost half. This system will be very memory extensive and will require a shorter access cycle for optimal performance.
- Computer for user interface.

  This will allow users to connect to the NLG portal and run the software.

Software Requirements

- Stanford CoreNLP

  The Stanford CoreNLP library is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution.

- SimpleNLG

  SimpleNLG is a library and not an application. SimpleNLG library is used to create a program that will generate grammatically correct sentences in English. The library consists of classes which allow the user to define the subject, the object, the verb and additional compliments to the sentence.

- MongoDB / NOSQL

  MongoDB (from humongous) is a cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure, having unstructured database architecture.

- Java JRE 6+

  Java JRE will allow us to run applets that provides user interface and running environment as the entire application is developed in Java. Java 2 Runtime Environment is essential to establish a connection between popular browsers and the Java platform. Java allows applications to be downloaded over a network and run within a guarded sandbox. Security restrictions are easily imposed on the sandbox.

- Eclipse

  Using Eclipse Integrated Development Environment for providing a development workspace and easy export and sharing of development files between team members along with building and publishing the final application. Eclipse uses plug-ins to provide all the functionality within and on top of the runtime system.

- Windows/Linux Operating Environment
  Provides and environment to sun the java application along with providing an interface to build and access the SAN network. This system should have access to the Internet, which is essential for connecting to the database.

CHAPTER 7

TEST CASES

CHAPTER 8

PROJECT TIME LINE

	REQUIREMENT ANALYSIS
	RESEARCH ON EXISTING SYSTEMS
	LOGIC DEVELOPMENT
	FAMILIARIZATION ON NATURAL LANGUAGEPROCESSING CONCEPTS
	LEARNING STANFORD NLP USUAGE
	LEARNING SIMPLENLG USUAGE
	JAVA LIBRARY IMPLEMENTATION FOR SENTENCE STRUCTURE AND POS TAGGING
	DATABASE FOR STORAGE OF INPUT, DICTIONARY AND LINKS CREATED BY THE JAVA PROGRAMS
	PPT PREPARATION
	WHITE BOOK DOCUMENTATION

	JAVA PROGRAM FOR FORMATION OF WORD LINKS AND SENTENCES
	TECHNICAL PAPER WRITING
	STATISTICAL LEARNING PROGRAM FOR MINING AND RETRIEVAL OF REQUIRED INFORMATION
	FINAL PROGRAM THAT ACCEPTS INPUT AND INTEGRATES ALL SECTORS TO OBTAIN FINAL OUTPUT
	PPT PREPARATION
	BLACK BOOK DOCUMENTATION

CHAPTER 9

TASK DISTRIBUTION

Responsibilities:

1. Java libraries for sentence structure and parts of speech:

   Sumer Shende
   Rishabh Vig

2. Database for storage of input, dictionary and links created by the java programs:

   Vineet Trivedi
   Vandita Shah

3. Documentation (White Book) and Presentation

Sumer Shende

Vandita Shah

Vineet Trivedi

Rishabh Vig

Time Frame:

August-November

Responsibilities:

1. Java program for formation of word links and sentences:

   Sumer Shende

2. Technical Paper

   Vandita Shah
   Vineet Trivedi

3. Statistical learning program for mining and retrieval of required information

   Rishabh Vig

4. Final program that accepts input and integrates all sectors to obtain final output

Sumer Shende

Vandita Shah

Vineet Trivedi

Rishabh Vig

5. Documentation (White Book) and Presentation

Sumer Shende

Vandita Shah

Vineet Trivedi

Rishabh Vig

Time Frame:

December-March

CHAPTER 10

CONCLUSION AND FUTURE WORK

Scribe can expand its data sources. It can be made capable enough to draw raw data from sources like RSS feeds of multiple website, social media plug roots, blogging stations, news tickers etc. By expanding its sources of data, the richness and quantity of data can be increased. For this, Scribe needs to be able to read text from different formats like XML, LateX along with grabbing the correct unformatted text from html documents.

Scribe has been implemented for producing articles for a cricket match. However its use is not bounded to only cricket. Scribe can easily learn how to write articles on other sports like football, hockey, baseball etc. For this Scribe needs to be fed training data in the form of articles from the respective sports and it will learn the structure and details of each sport from the training data itself. With minimal changes made to the Scribe it will be ready to use for any other sport. However the most important use of Scribe is the coverage of natural disasters. As stated earlier one of the key problems that this system was designed to tackle and eliminate was human delay in communication. During natural disasters chances of human delay are high and the need to communicate the disaster and its consequences is of paramount importance. Abnormal readings from instruments like seismometers, Saffir-Simpson scale and stream gauges indicating a natural disaster like earthquake, hurricane or flood respectively should be communicated to Scribe. Scribe can then write a piece informing the people of a natural disaster. Scribe can be one of the fastest modes of communicating the occurrence of a natural disaster, the area it has occurred in and its possible impact.

Incorporating the above suggestions in scribe will make the proposed system a multifaceted all-purpose journalist capable of writing an article on any given topic.

CHAPTER 11

REFERENCES

- Abhimanyu Chopra, Abhinav Prashar, and Chandresh Sain, 2013, “Natural Language Processing”, International Journal of Technology Enhancements and Emerging Engineering Research.
- E. Cambria and B. White, 2014, “Jumping NLP Curves: A Review of Natural Language Processing Research”, IEEE Computational Intelligence Magazine.
- J. M. Huerta and D. Lubensky, 2003, “Graph-based representation and techniques for NLU application development”, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03).
- N. Ito and M. Hagiwara, 2011, “Natural language generation using automatically constructed lexical resources” The 2011 International Joint Conference on Neural Networks (IJCNN).
- Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard and David McClosky, 2014, “The Stanford CoreNLP Natural Language Processing Toolkit”,

Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations.

- Zachary Parker, Scott Poe and Susan V. Vrbsky, 2013, “Comparing NoSQL MongoDB to an SQL DB”, Proceedings of the 51st ACM Southeast Conference.
- Veronika Abramova and Jorge Bernardino, 2013, “NoSQL databases: MongoDB vs cassandra”, Proceedings of the International C* Conference on Computer Science and Software Engineering.
- Albert Gatt and Ehud Reiter, 2009, “SimpleNLG: A realisation engine for practical application”, Proceedings of the 12th European Workshop on Natural Language Generation.
- A Robust Human-Robot Communication System Using Natural Language for HARMS (The 12th International Conference on Mobile Systems and Pervasive Computing, MobiSPC 2015)
- SimpleNLG: A realisation engine for practical applications. (Proceedings of the 12th European Workshop on Natural Language Generation)
- Collaborative Writing Support Tools on the Cloud (IEEE Journal on Learning Technologies, VOL. 4, NO. 1, March 2011)
- Discourse strategies for generating natural-language text(Artificial Intelligence (Elsevier, Volume 27, Issue 1))

- Natural Language Generation as Incremental Planning Under Uncertainty: Adaptive Information Presentation for Statistical Dialogue Systems (IEEE/ACM Transactions on Audio Speech And Language Processing, VOL. 22, NO. 5, May 2014)
- Big Data and the SP Theory of Intelligence (Digital Object Identifier 10.1109/ACCESS.2014.2315297)
- Using Wikipedia and Conceptual Graph Structures to Generate Questions for Academic Writing Support (IEEE Transactions On Learning Technologies, VOL. 5, NO. 3, July-September 2012)

CHAPTER 12

APPENDIX

Rough notes:

Add intro mai motivation

Requirement analysis karo

Add test cases and project time line

Add distribution of work