TEXT SUMMARIZER AND HEADLINE GENERATOR

Table of Contents

1. INTRODUCTION

1.1 Purpose

- This document contains the system requirements for News Headline Generation Project.
- It includes descriptions of the functions and the specifications of the project.

This document is intended for:

- Developers who intend to develop and extend headline generation project.
- Faculty of PEC

1.2 Scope

- News Headline Generation System is a software which generates headline to input news article. Headline is an abstract one line which conveys overview of article.
- The system will produce summary for article given as input by user and feedback is taken from user and changes are made in summary according to feedback and headline is generated from summary and feedback.
- In contrast to human-generated newspaper headlines, our approach produces informative abstracts, describing the main theme or event of the newspaper article (e.g., “Wide Gap Between 3 Best Players”).
- The software can be used by general public as software extracts important extract the most relevant piece of information out of a tons of data in form of headline-style abstracts.

1.3 Definitions, Acronyms, and Abbreviations

Table – 1 Terms and Definition

Terms	Definition
Tokenization	Tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens
Stemming	Stemming usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes.
Natural Language Processing	Natural language processing is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction.

Lemmatization	Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma .
POS Tagger	A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'.
Stop word	Stop word is a commonly used word (such as "the") that has been programmed to ignore
Parsing	A natural language parser is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb
Machine Learning	Machine learning is the subfield of computer science that gives computers the ability to learn without being explicitly programmed. Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data driven predictions or decisions
Python	Python is a simple yet powerful programming language with excellent functionality for processing linguistic data. It has good string-handling functionality.
TAG	A unique, persistent identifier contained in a PLanguage statement
GIST	A short, simple description of the concept contained in a PLanguage statement
SCALE	The scale of measure used by the requirement contained in a PLanguage statement
METER	The process or device used to establish location on a SCALE contained in a PLanguage statement
MUST	The minimum level required to avoid failure contained in a PLanguage statement
PLAN	The level at which good success can be claimed contained in a PLanguage statement
WISH	A desirable level of achievement that may not be attainable through available means contained in a PLanguage statement
DEFINED	The official definition of a term contained in a PLanguage statement

1.4 References

1. IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.
2. Feldt R,”re_lecture5b_100914”, unpublished.

1.5 Overview

The remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product. The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences. The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen. The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

2. OVERALL DESCRIPTION

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

2.1 Product perspective

This system will consist of a software which needs to be installed first on the device for its use. The software will be used to summarize the text or an article entered by user and will also generate the most likely headline for the article or text. User will have to enter the text first and then needs to select the length of the summary needed and the software will generate the summary and headline as per the requirements of the user.

The software will need to analyze the scores and database in order to generate the headline as well the summary (figure 1). The software will provide the summary and the headline but the user can change the headline as well as the summary generated if it is not as per his needs. The functionality to provide multiple summaries and the headlines are included in the software.

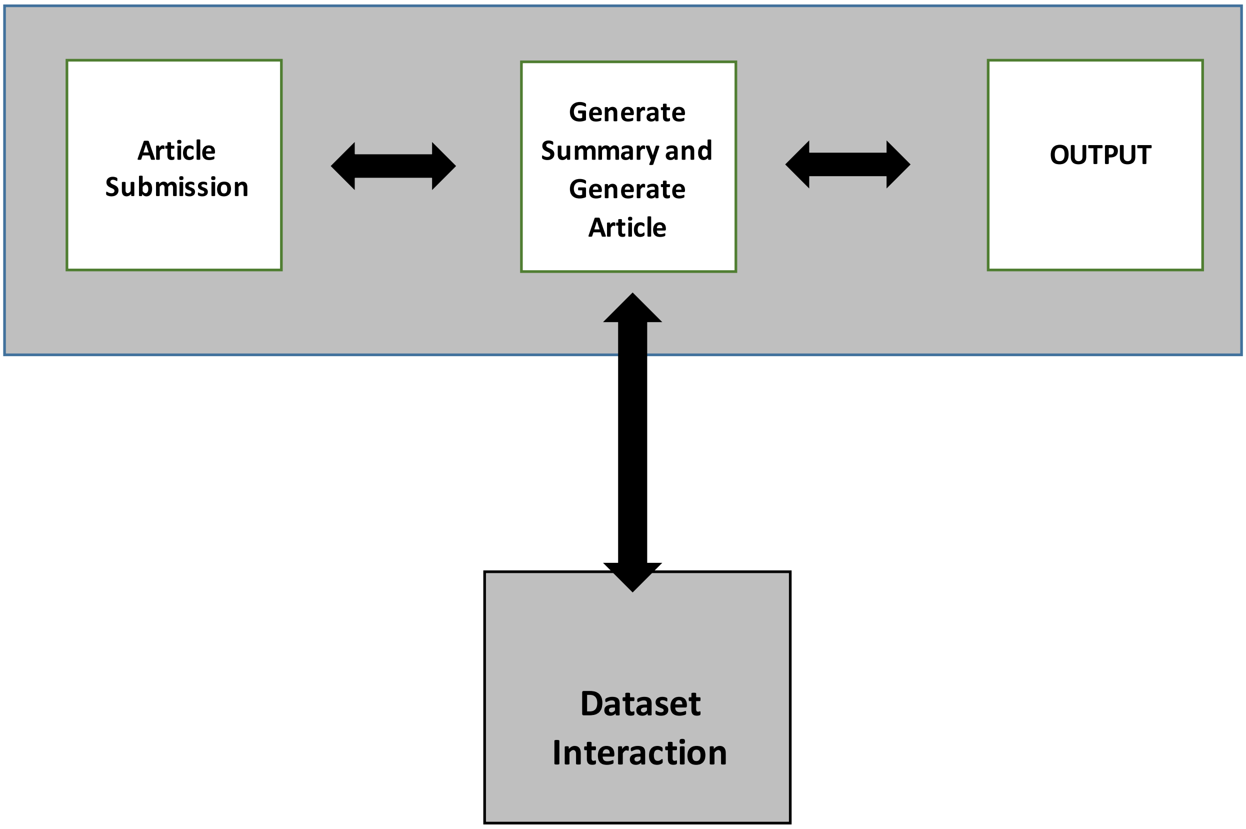

Figure 1

2.1.1 Software Interface

- Text Summarizer, generating appropriate summaries and headlines for the submitted articles
- Trained Datasets used for fetching the information and are used to assign scores in order to get the appropriate output

2.1.2 Hardware Interface

- Any Display device (mobile, Tablets, PC's) and mobiles with android version 4.4 or above

2.1.3 Memory Constraints

Since this is a data-centric product as a lot of data is needed to be analyzed it will need somewhere to store the data. For that, a database will be used. The software will communicate with the database for storing results and fetching the scores. To avoi d problems with overloading the operating system the software is only allowed to use 60 megabytes of memory while running the software. The maximum amount of hard drive space is also 60 megabytes.

2.1.4 Operations

2.1.4.1 Various operations in the software

1. Submit an article
2. Select the option:
   a. Generate Summary
   b. Generate Headline
3. Save the output

2.1.4.2 Back Up and recovery options

The backup will be provided in form of history which will store the activities of the user within 24 hours. The user will be able to access all the activities and can recover the lost data within 24 hours.

2.1.5 Site Adaptation Requirements

For this system we need to have databases which will modify the data and update the dataset simultaneously. The software will require sufficient RAM to operate as all the database will be needed to fetch the correct results.
For this system we are using nltk corpus and this corpus will be modify simultaneously as per the results given by it.

2.2 Product functions

With the software, the user will be able to shorten the text and can generate automated topic headline. The interface will be such that user will ask to enter an article and asked to choose the options which are to generate summary and to generate headline.
The result of the input can be saved in the form of word document or in the form of pdf as per the user requirement.
So basically, the main functionality of the software is only to give user an accurate and precise summary and to make an automated headline.

2.3 User Characteristics

The end users of the software can only interact with the software to get headline of article. The customers are not expected to have a high educational and proficiency level or technical expertise. User interface is available in English. The user has to be able to input news article, enter feedback for the summary generated by system and validate summary.

2.4 Constraints

The software must run on a system that has support for 60MB memory. The system must support access to directory file structure as software needs to maintain database and modify it on a regular interval. The system uses different interfaces of nltk so the system must use the language supported by installed version of nltk.

2.5 Assumptions and Dependencies

One assumption for the software is that computer has enough performance. The system has high computational requirements and thus computer which uses system must have high performance RAM.
Another assumption is that system supports the nltk software and the language which supports nltk.

2.6 Apportioning of Requirements

In case the project gets delayed, the first increment must have at least the functionality of text summarization. In next increment, headline is generated out of the summary generated. In next and final increment, feedback is taken from user for summary and is incorporated in future results.

3. SPECIFIC REQUIREMENTS

3.1 External interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

2.1.1 System Interfaces

The news headline generator software will be interacting with various interfaces tokenizer, POS tagger, lemmatizer, stemmer, stop words filtering and synonyms identification.

1. **Tokenization**: In lexical analysis, **tokenization** is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing or text mining. For this process, we will be using the python library- nltk. Tokenizer (word tokenizer).
2. **Stemming:** For grammatical reasons, documents are going to use different forms of a word, such as *organize*, *organizes*, and *organizing*. Additionally, there are families of derivationally related words with similar meanings, such as *democracy*, *democratic*, and *democratization*. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set.
3. **Lemmatization:** usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the ***lemm**a*. If confronted with the token *saw*, stemming might return just *s*, whereas lemmatization would attempt to return either *see* or *saw* depending on whether the use of the token was as a verb or a noun
4. **Parser:** Analyze (a string or text) into logical syntactic components. (add example here). We will be using the nltk parser for this purpose. By this time, we have the tokens ready and each one of them have a specific POS tag.

5. **Stop words filtration and synonyms identification:** The synonyms identification is done by using the dictionary word net and stop words filtration will be performed by using a library in nltk.

2.1.2 User Interfaces

A first-time user of the mobile application should see the long text box and an option to choose file see Figure 2. If the user has a file to be summarized, then he can directly choose the file or can enter the text in the text box.

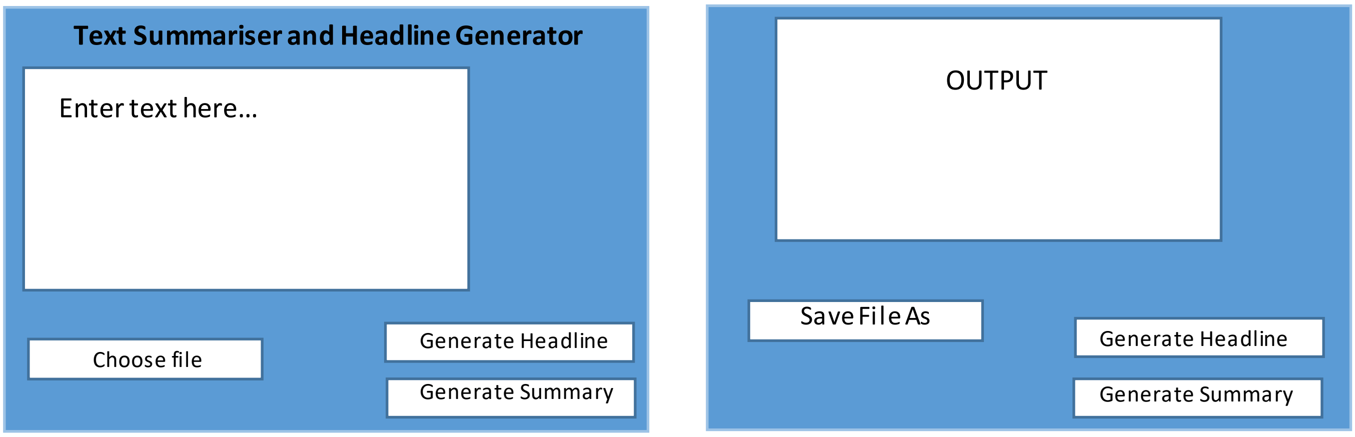

Figure 2

After entering the text or the file user should select the option whether to generate summary or to generate headline or both.
The output window will appear containing the summarized text or a headline or both. User can select these options again in order to generate different summary or headline. By selecting these options again and again user will get different output every time and he can select the most suitable one as per his requirements.

2.1.3 Hardware Interfaces

Since the software does not have any designated hardware, it does not have any direct hardware interfaces. The software will have to be installed in the device first for its use, all the databases will automatically get installed with the software.

2.1.4 Software interfaces

The software communicates with the trained datasets(database) in order to fetch the probabilities of the occurrence of words and to fetch the scores that are to be assigned to the sentences to be included in the summaries. The communication between the database and the software is of operations concerning both reading and modifying the data.

2.1.5 Communications interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for the software.

3.2 Functional Requirements

This section includes the requirements that specify all the fundamental actions of the software system.

3.2.1 User Class

ID: FR1
TITLE: Install the Text summarizer and headline Generator
DESC A user should be able to download and install the software through either an application store or similar service on the web. The application should be free to download.
RAT: In order to install the software
DEP: None

ID: FR2
TITLE: User Input
DESC: After downloading the user will enter the input as a text or any text file which is to be summarized as well as generate headline for the text.
RAT: In order for the user to enter input.
DEP: FR1

ID:FR3
TITLE: Selecting the summary and headline
DESC: This enables the user to select the the appropriate summary as per user requirements as the software has the functionality to generate the multiple summaries and headlines for the user.
RAT: In order to select the appropriate summary and headline.
DEP: FR1

ID: FR4
TITLE: Saving Output
DESC: This will help user to save output in the specific format either in .docx or in pdf as per user need.
RAT: In order to save the results
DEP: FR1

ID: FR5
TITLE: Update Data Set
DESC: Data Set needs to be updated simultaneously with the generated output. We need to train the data set constantly during the lifetime of the system.
RAT: In order to modify and update the database
DEP: FR1

ID: FR6
TITLE: Download and notify users of new releases
DESC: When a new/updated version or release of the software is released, the user should check for these manually.
RAT: In order for user to download updated release
DEP: FR1

ID: FR7
TITLE: Recover output
DESC: It will enable the user to recover the lost data within last 24 hours. This will act as a backup for the user in case of any errors or system failure.
RAT: In order to retrieve lost data.
DEP: FR1

3.3 Performance requirements

The requirements in this section provide a detailed specification of the use r interaction with the software and measurements placed on the system performance.

Static Requirements: -

a) **No. of simultaneous users to be supported**: In present version, only one user is supported. In latest versions, multiple users can use the software on the same system through different account ID's.

b) **Amount of information to be handled:** Information is already stored in the trained dataset which needs to be modified simultaneously with every output returned by software.

c) **Type of information to be handled:** The input will be only in the form of text which will be then processed by the software in order to give the accurate output.

Dynamic Requirements: -

a) **Amount of Data to be processed:** The data to be processed will be in the form of text only and there is no limit on the size of article entered by user. The user can enter any amount of data but the generated output will be restricted to some particular number of characters only.

b) **System Dependability:** If the system failure occurs, then user will be informed. The backup will be generated for the same.

c) **Response Time:** The response time will depend upon the size of article submitted. The larger the article, more time will be taken to process the text and generate the output.

3.4 Logical Database Requirements

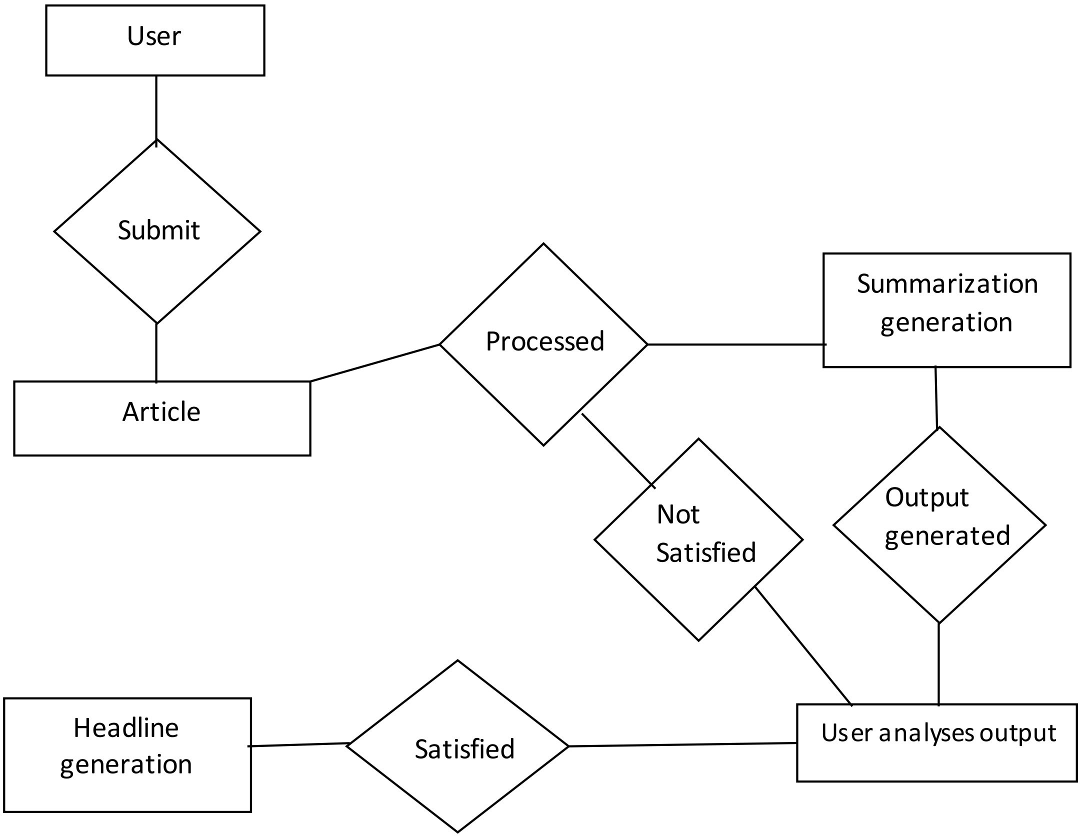

Figure 3- Logical Structure of the Article Manager Data

The data descriptions of each of these data entities is as follows:

User Data Entity

Data Item	**Type**	**Description**	**Comment**
Name	Text	Name of principle author	
Article	Pointer	Article entity	May be several

Summary Generation Data Entity

Data Item	Type	Description	Comment
Name	Algorithms	Various Machine learning techniques	
ID	Integer	ID number of Output generated	Used as key in Database
Article	Pointer	Article entity of	May be several
Returned	Summarized Article	Article summary is returned	

Headline Generation Data Entity

Data Item	Type	Description	Comment
Article	Pointer	Article entity	
Data Sent	Textual Data	Data sent to data set for modification	
Returned	Headline	Headline for the article returned	

Article Data Entity

Data Item	Type	Description	Comment
Name	Text	Name of Article	
User	Pointer	User entity	Name of user
Contents	Text	Body of article	Contains Abstract as first paragraph.
Category	Text	Area of content	May be several

3.5 Design Constraints

This section includes the design constraints on the software caused by the hardware.

3.5.1 Hard drive space

TAG: Hard DriveSpace

GIST: Hard drive space.

SCALE: The application's need of hard drive space.

METER: MB.

MUST: No more than 60 MB.

PLAN: No more than 50 MB.

WISH: No more than 40 MB.

MB: DEFINED: Megabyte

3.5.2 RAM memory

TAG: RAM Memory

GIST: The amount of RAM memory required by the application

SCALE: GB.

METER: Observations done from the performance log during testing

MUST: No more than 8 GB.

PLAN: No more than 4 GB

WISH: No more than 4 GB

MB: DEFINED: Gigabyte.

3.6 Software System Attributes

3.6.1 Reliability

This software will be developed with machine learning, feature engineering and natural language processing techniques. So, in this step there is no certain reliable percentage that is measurable. The results are based on machine learning techniques and fee dback provided by user. The feedback provided gives the direction to flow of estimating important keywords, thus results of system also depends upon input given by user. The system has given satisfactory results on test cases, however satisfaction is a subjective matter. Humour, sarcasm, pun etc. such elements are missing from headline.

3.6.2 Usability

The system should be easy to use. The user should reach the summarized text with one button press. Because one of the software's features is timesaving. The user should be able to give feedback easily and change in summary should clearly depict inclusion of feedback. The headline generated should be available with one click of user.

3.6.4 Performance

Calculation time and response time should be as little as possible, because one of the software's features is timesaving. Whole cycle of summarizing an article will depend upon size of article nd should not take too long. Response time should be very low in case of taking feedback from user. The time taken to produce headline from generated summary should be minimal and should decrease with increased usage. The quality of headline shall also increase as the usage of software increase. After 60MB usage of memory by system, it shall start replacing data in database rather than adding. So, algorithm used to replace data in database should be efficient enough so that files with present context are not deleted and quality of headline generated should not be affected by replacement.

3.6.5 Portability

The system has nltk components as one of the interfaces. Nltk has different compatible forms in Linux, Windows and Mac thus different packages of software needs to be released for different OS system.