

**Informatics Institute of Technology**

in collaboration with

**University of Westminster, UK**

**CreativeAid!**

**“Autonomous Creative Title Generation and Creative Text  
Identification Framework”**

A dissertation by

**Ahamed Shimak (2015312)**

**Supervised By**

**Mr. Saman Hettiarachchi**

Submitted in partial fulfillment of the requirements for the

BEng (Hons) Software Engineering degree

Department of Computing

May 2019

© The copyright for this project and all its associated products resides with Informatics  
Institute of Technology.

## Declaration

I hereby certify that this project report and all the artifacts associated with it are my own work and neither it has been submitted before nor is currently being submitted for any degree program.

Full Name of Student: Mohamed Naushad Ahamed Shimak

Registration Number: 2015312

Signature: \_\_\_\_\_

Date: \_\_\_\_\_

## Abstract

Creativity is an art which aims at inducing certain set of beliefs in the target audience. In digital marketing and media, when promoting article or post creative titles are often composed to attract audience towards read the post or article. Such titles can be catchy, attractive persuasive etc... Business uses this technique to attract and reach many audiences. Creating attractive titles often depend on humans. They can't be creative every time hence they depend on different tools to boost creativity. Existing tools provide analysis solution they lack in creativity-based solution.

In this project we present CreativeAid! – a creative title generation and creative sentence identification framework. It helps to build application to generate creative title automatically by just providing title or title's contents and templates. Moreover, the framework also provides feature to identify creative sentences in given sentences. With these features' developer can setup effective creative title generation application.

Proposed solution tested thoroughly under different conditions and the framework was evaluated by evaluators of various domains. Eventually, the test results proved that the analysis, design and implementation was approved in an effective and in an efficient manner.

### Keywords

Natural Language Generation, Natural Language Understanding, Machine Learning, Text Processing, Catchy Sentence Generation

### Subject descriptors

• **Computing methodologies~Artificial intelligence** • **Computing methodologies~Natural language generation**

## **Acknowledgment**

A large number of resources in terms of time and finances were put into the conducting and completion of this particular. I would like to convey my utmost appreciation to my supervisor Mr. Saman Hettiarachchi, whose contribution in stimulating feedbacks and support, helped me to coordinate my project. It was very crucial in overcoming numerous obstacles.

Also, I would like to express my gratitude to Mr. Kaneeka Vidanage (Module Leader). I would like to express my heartfelt appreciation, to my loving parents for their continued support and encourage. My friends who have helped me whenever I needed a helping hand, the academic and non-academic staff of IIT who provided me with support throughout the degree program, I am indeed grateful to you all.

## Table of Contents

Declaration.....	i
Abstract.....	ii
Acknowledgment.....	iii
List of Figures .....	x
List of Tables .....	xii
List of Equations .....	xiii
List of Abbreviations.....	xiv
1. Introduction.....	1
1.1 Chapter Overview .....	1
1.2 Project background.....	1
1.3 Problem Domain .....	2
1.4 Research Problem .....	3
1.6 Research Question .....	4
1.7 Project Aim .....	4
1.8 Motivation .....	4
1.9 Objectives.....	4
1.9.1 Research Objectives.....	4
1.9.2 Personal Objectives .....	5
1.10 Project Scope .....	6
1.11 Resource Requirements .....	6
1.12 Rich Picture of the project.....	7
1.13 Chapter Summary.....	8
2. Project Management.....	9
2.1 Chapter Overview .....	9
2.2 Data Gathering Approaches.....	9
2.3 Project Management Methodology.....	9
2.3.1 Time allocation.....	10
2.3.2 Constrains .....	10

2.4	Potential Risks and Mitigation Plan .....	10
2.5	Research Methodology .....	12
2.6	Development Methodology .....	12
2.7	Design Methodology .....	13
2.8	Evaluation Methodology.....	13
2.9	Work Breakdown Structure .....	14
2.10	Activity Schedule .....	14
2.11	Chapter Summary.....	14
3.	Literature Review .....	15
3.1	Chapter overview.....	15
3.2	Automatic title generation system .....	15
3.3	Classification of title types.....	15
3.4	Related Studies .....	16
3.4.1	Title generation system related studies.....	16
3.4.2	Creative text identification related studies.....	19
3.4.3	Creative sentence generation related studies .....	20
3.5	Creative sentence identification approaches.....	21
3.6	Creative title generation approaches.....	24
3.6.1	Semantic words .....	24
3.6.2	Similarity measures.....	25
3.7	Title generation approaches.....	25
3.7.1	Abstractive Text Summarization .....	25
3.7.2	Extractive Summarization .....	26
3.8	Chapter Summary.....	29
4.	Requirement Specification .....	30
4.1	Chapter Overview .....	30
4.2	Stakeholders Analysis.....	30
4.2.1	Stakeholders Onion Model.....	30
4.2.2	Stakeholders roles and description .....	31

4.3	Requirement Elicitation .....	32
4.3.1	Literature survey .....	34
4.3.2	Interviews .....	34
4.3.3	Online Questionnaire .....	35
4.3.4	Analysis of technical questionnaire .....	36
4.3.5	Findings derived from the Requirement Elicitation .....	38
4.4	Analysis & Design Methodologies .....	39
4.4.1	Design methodology .....	39
4.4.2	Modelling Language.....	39
4.5	Use Case Diagram.....	40
4.5.1	Use Case Descriptions .....	41
4.6	Activity Diagram.....	43
4.6.1	Activity Diagram for identification part of framework.....	43
4.6.2	Activity diagram of creative title generation .....	44
4.7	Functional Requirements .....	45
4.8	Non-Functional Requirements .....	46
4.9	Scope Refinement .....	47
4.10	Chapter Summary.....	47
5.	Design Specification .....	48
5.1	Chapter Overview .....	48
5.2	Design Goals .....	48
5.3	Architecture Styles.....	48
5.4	High Level Architecture of the Framework .....	50
5.4.1	Modules of framework.....	50
5.5	Low level design models.....	51
5.5.1	Class Diagram .....	52
5.5.2	Sequence Diagram .....	54
5.5.3	Context Diagram.....	54
5.6	Chapter Summary.....	55

6.	Implementation .....	56
6.1	Chapter Overview .....	56
6.2	Technology selection .....	56
6.2.1	Programming language selection.....	56
6.2.2	Libraries Selection .....	56
6.2.3	Word Embeddings Selection.....	57
6.2.4	Clustering library selection.....	57
6.3	Tools selection.....	58
6.3.1	IDE selection.....	58
6.3.2	Version Control System selection .....	58
6.4	Data and Models preparation.....	58
6.4.1	Clustering model.....	58
6.4.2	Selectional Preference data.....	60
6.4.3	Keywords selection model .....	61
6.5	Technology stack.....	61
6.6	Implementation of Functional requirements .....	62
6.6.1	Receive input as raw sentences or corpus format.....	62
6.6.2	Process documents .....	62
6.6.3	Creative sentence identification .....	63
6.6.4	Identify alias and pseudonym names .....	64
6.6.5	Generate Title.....	65
6.6.6	Creative text templates identification for creative title.....	66
6.6.7	Find important words in title and templates.....	68
6.6.8	Generating creative title .....	69
6.7	Problems Encountered .....	71
6.8	Chapter summary .....	71
7.	Testing.....	72
7.1	Chapter Overview .....	72
7.2	Testing Goals.....	72



7.3	Test application description.....	72
7.4	Testing criteria .....	72
7.4.1	Functional Testing .....	72
7.4.2	Non-Functional Testing.....	73
7.5	Selection of Testing Framework.....	73
7.6	Test execution and results .....	73
7.6.1	Unit Testing .....	74
7.6.2	Module and Integrating Testing.....	74
7.6.3	Accuracy Testing .....	76
7.6.4	Performance Testing.....	80
7.7	Remarks and Limitation of testing .....	82
7.8	Chapter Summary.....	83
8.	Evaluation .....	84
8.1	Chapter Overview .....	84
8.2	Evaluation Goals.....	84
8.2.1	Evaluation of the concepts.....	84
8.2.2	Evaluation of the Technical Aspects .....	84
8.2.3	Evaluation of the Usefulness and Impact of framework.....	84
8.3	Selection of Evaluators .....	85
8.4	Evaluation Methodology.....	85
8.5	Execution of evaluation methods .....	85
8.5.1	Evaluation of the concepts.....	86
8.5.2	Evaluation of Technical Aspects .....	87
8.5.3	Evaluation of Usefulness and Impact .....	90
8.6	Benchmarking.....	92
8.7	Research Questions and Answers .....	93
8.8	Critical Evaluation .....	93
8.8.1	Evaluation on concept.....	94
8.8.2	Evaluation on technical aspects.....	94

8.8.3	Evaluation of the Usefulness and Impact of framework.....	95
8.9	Chapter Summary.....	96
9.	Conclusion.....	97
9.1	Chapter Overview.....	97
9.2	Achievements of Objectives.....	97
9.3	Milestones and Deliverables.....	98
9.4	Achievement of Requirements.....	99
9.4.1	Achievements of functional requirements.....	99
9.4.2	Achievements of non-functional requirements.....	100
9.5	Achievement of Aim.....	100
9.6	Limitations of the Project.....	101
9.7	Learning Outcomes.....	101
9.8	Future Enhancements.....	101
9.9	Problems and Challenges Faced.....	102
9.10	Concluding Remarks.....	102
	References.....	A
Appendix A	Word Breakdown Structure.....	F
Appendix B	Interviewees.....	G
Appendix C	Use Case Descriptions.....	G
Appendix D	Sequence Diagrams.....	N
Appendix E	Evaluators.....	Q
Appendix F	Unit Test cases.....	R
Appendix G	Module Integration Test.....	X
Appendix H	Title template word pair.....	Z
Appendix I	Random Templates.....	Z
Appendix J	Finding keywords.....	BB
Appendix K	Activity Schedule.....	DD

## List of Figures

Figure 1.1 - (Simon Kemp, 2018) research report .....	1
Figure 1.2 - Rich Picture .....	7
Figure 3.2 - Sample graph build for TextRank extractive summary as depicted in (Mihalcea and Tarau, 2004). Each node represents a sentence.....	27
Figure 4.1 - Onion model .....	31
Figure 4.2 - Better programming language for NLP interview question & answers .....	36
Figure 4.3 - Provide method for developers to implement interview question and answer ...	37
Figure 4.4 - Provide summary report interview question and answer .....	37
Figure 4.5 - Important requirement for framework interview question and answer .....	38
Figure 4.6 - Use case diagram .....	40
Figure 4.7 - Activity diagram for identification .....	43
Figure 4.8 - Activity diagram for generation.....	44
Figure 5.1 - High Level Architecture Diagram.....	50
Figure 5.2 - Class diagram.....	52
Figure 5.3 - Sequence diagram for creative text identification .....	54
Figure 5.4 - Context diagram.....	55
Figure 6.1 - IDE comparison (Slant, 2019) .....	58
Figure 6.2 - Verb cluster groups.....	59
Figure 6.3 - Noun cluster groups.....	59
Figure 6.4 - Selection preference data gathering.....	60
Figure 6.5 - Selection preference word pair normalizing.....	60
Figure 6.6 - Technology stack .....	61
Figure 6.7 - Corpus reader implementation .....	62
Figure 6.8 - Clean sentences implementation .....	63
Figure 6.9 - Identify unique implementation.....	63
Figure 6.10 - Process candidate implementation.....	63
Figure 6.11 - spaCy pipeline example (Explosion AI, 2019) .....	63
Figure 6.12 - Creative text identifier pipe implementation.....	64
Figure 6.13 - Vectorizing implementation .....	64
Figure 6.14 - Generate title implementation .....	65
Figure 6.15 - Sentence similarity implementation .....	65
Figure 6.16 - Similarity matrix implementation.....	66
Figure 6.17 - Search candidate implementation .....	66
Figure 6.18 - Title sentence similarity implementation .....	66
Figure 6.19 - Keyword score implementation .....	67

Figure 6.20 - Selectional preference strength calculation implementation .....	67
Figure 6.21 - Noun probability calculation .....	67
Figure 6.22 – Conditional probability calculation .....	68
Figure 6.23 - Selectional Association calculation.....	68
Figure 6.24 - Important keyword implementation.....	69
Figure 6.25 - Substitute words implementation.....	69
Figure 6.26 - Creative title generation implementation .....	70
Figure 6.27 - Substitute words implementation.....	70
Figure 7.1 - Framework testing modules .....	75
Figure 7.2 - Creative text identification performance .....	82
Figure 8.1 - Developer friendliness rating evaluation.....	90
Figure 8.2 - Developer friendliness rating reason evaluation .....	90
Figure 8.3 - Prototype aspects evaluation .....	91
Figure 8.4 - Aspects of framework evaluation .....	91

## List of Tables

Table 1.1 - Research objectives .....	5
Table 1.2 - Personal objectives .....	5
Table 2.1 - Tasks of the project and time allocation .....	10
Table 2.2 - Risks and Constraints of System with mitigation plan.....	12
Table 4.1 – Stakeholder roles and description .....	32
Table 4.2 - Requirement elicitation methods .....	34
Table 4.3 - Analysis of Interview .....	35
Table 4.4 - Findings derived from requirement elicitation .....	39
Table 4.5 - Use case description .....	42
Table 4.6 - Functional requirements.....	46
Table 4.7 - Non-functional requirements .....	47
Table 5.1 - Class diagram description .....	53
Table 6.1 - Library selection comparison.....	57
Table 7.1 - Unit test results summary .....	74
Table 7.2 - Module integration test results.....	76
Table 7.3 - Summary results of finding keywords .....	79
Table 7.4 - Results of test finding suitable template .....	80
Table 7.5 - Creative title generation performance testing .....	81
Table 8.1 - Scope and depth evaluation .....	86
Table 8.2 - Need for creative identification evaluation .....	86
Table 8.3 - Use and impact in the field evaluation .....	87
Table 8.4 - Framework architecture and design evaluation .....	88
Table 8.5 – Coding practices evaluation.....	89
Table 8.6 - Tools and technologies evaluation .....	89
Table 8.7 - Developer friendliness evaluation.....	90
Table 8.8 - Features of framework evaluation .....	91
Table 8.9 - Accuracy of framework evaluation.....	92
Table 8.10 - Comparison of features .....	93
Table 9.1 - Achievements of objectives.....	98
Table 9.2 - Milestones deliverables .....	99
Table 9.3 - Achievements of functional requirements.....	100
Table 9.4 - Achievements of non-functional requirements.....	100
Table 9.5 - Future enhancements .....	102

## List of Equations

Equation 3.1 - SPS calculation .....	23
Equation 3.2 - SA calculation .....	24
Equation 7.1 – Creative text identification accuracy test.....	77
Equation 7.2 - Accuracy of finding suitable keywords.....	78
Equation 7.3 - Finding suitable template accuracy .....	79
Equation 7.4 - Performance test case .....	81

## List of Abbreviations

Abbreviation	Denotation
API	Application Programming Interface
FR	Functional Requirement
IDE	Integrated Development Environment
IT	Information Technology
ML	Machine Learning
NA	Not Available
NER	Name Entity Recognition
NFR	Non-Functional Requirement
NLG	Natural Language Generation
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NLU	Natural Language Understanding
OOA	Object Oriented Analysis
RAM	Random Access Memory
SA	Selectional Association
SE	Software Engineering
SPS	Selectional Preference Strength
SysML	Systems Modelling Language
UML	Unified Modeling Language
UML	Unified Modeling Language
VCS	Version Control System
WBS	Work Breakdown Structure

# **Chapter 1: Introduction**



# 1. Introduction

## 1.1 Chapter Overview

This chapter discusses the project fundamentals and provide outline about rest of the chapters. In details it gives an idea about the project background, problem, motivation, objectives and aim to the do the research project.

Moreover, the chapter describes the methodologies, scope, requirements are analyzed, and suitable approaches were picked. Furthermore, rich picture, WBS and activity schedule are illustrated.

## 1.2 Project background

The world we live in is always in change, everything has an amazing speed and the information has become crucial. Internet is growing as an information platform. With the increase of smart devices usage people can access information within seconds.



Figure 1.1 - (Simon Kemp, 2018) research report

It is not wrong to say role of media is huge. People rely on the information given by these outlets. With the rise of posts in media it is very competitive to get readers attention. Most of the time readers only read the title of the post and if they attracted, they will get into the main story otherwise they skip to next post. Studies says that "80% of readers never make it past the headline" and according to some sources, on average, eight out of ten people will read headline copy, but only two out of ten will read the rest (Nathan, 2013). Here the title is the

key element of content writing. The title's role is highlighted by the fact that one can find it on the top of the article.

A good title is a difference between encouraging someone to read the story and forgetting to read something else. The writers know how to use the right words and the textual and grammatical methods to give the information in different forms. They invest more time to come up with great headlines for every article because the more attractive the title the more users read. Attractive title increases user engagement, readability, conversions, SEO (Search Engine Optimization), social share and even click-through rate.

Creating a great title hasn't been the same way all these years. It changes like trends in fashion they come and go, changing and shifting with popular sentiment and the main objective behind the title is that the readers want to be entertained, informed & promise them in return something when reading a post, the title of the post should be guarantee that. A great title comes with many advantages and boring title will make outflow and make readers leave the platform, loose encouragement & reduce interest in page.

Making great title to reach readers is responsible of writers so they have to aware of social trending, right linguistic styles, different sources for detailed news. With the competition to reach readers, writers use tools to get insight about title and source. Some of them are emotional marketing, value analyzers, title analyzers, title generators, SEO analyzers, social media analyzers.

Among them title generators playing an important role. Title generating tools generate title from given contents considering different aspects. Current title generation tools consider more aspects to make great title.

### **1.3 Problem Domain**

Prior title generating work generate a very short summary of article and produce it as title (Alfonseca, Pighin and Garrido, 2013). The rise of NLP and Machine learning made huge impact on headline generation. Many researches attempt to make great title generating approaches considering different aspects; produce title in informative way (Alfonseca, Pighin and Garrido, 2013), Title generation avoiding Click-baits, headlines with styled manner (Shu *et al.*, 2015), Generate title that expected to be widely shared on social media (Szymanski, Orellana-Rodriguez and Keane, 2017).

Informative title is not enough to get users attention. Piotrkowicz (2017) says "the fundamental issue for people who writes title is how to more effectively attract readers. This question applies not only to journalists (who routinely write title and have received appropriate training), but to all authors of the myriads of user-generated content from blog posts to videos, which

feature a title". Title generating system researches that made to reach target audience considers different approaches. System identifies different aspects like what's trending in social media, catchy title, influential words in SEO. One of the good practices to make user notice title is using creative words and phrases such as idioms, catchy words and expressions and metaphors.

*"End of the dream for migrant builder who lost leg in cave-in"*

For an instance, In above title Lucy (2007) the journalist uses a creative words attract readers and the mean time provide information about the contents. Author wants to emphasize how huge the migrant problem is. "End of the dream" is used in a figurative way because when you have a serious problem, you feel like you have no hope or luck. In this title comparison between the migrant's situation and "End of the dream" phrase emerge feeling that reader can understand. We see a young migrant which crashed when we had an accident at work and the end of a dream, when the reality is not so easy to take. The poor migrant builder can't work no more and help his family because, unfortunately, he had an accident which ruined his life forever. This example explains the creative word use in title and it is important for writers to find suitable creative text.

"Creative word and phrases are employed to raise consumer awareness and manipulate the decoding and comprehension processes of target audiences"el (McQuarrie and Mick, 1992). They are being used in different areas marketing & advertising to reach customers. As Philips and McQuarrie (2009) say observe, research into creative text type effects typically compares the impact of creative text containing/not containing creative text, attributing any differences to style. Results are interpreted as demonstrating the importance of style factors relative to content. Creative text is a key feature for creative writing and automating this process to identifying and generating is a huge challenge.

## **1.4 Research Problem**

***Use Selectional Preference approach to identify creative language in text and generate creative title using creative language.***

Previous research approaches have limited title generation to informative title and research related to creative text generation limited to use own data set which couldn't be modified.

## 1.6 Research Question

The following are the main three research questions for the project.

- a. What is the best approach to identify creative text in text?
- b. Can system generate title with creative words and phrases; in a manner What is the best approach to generate creative title?
- c. What is the best way to find the suitable creative sentence for title?

## 1.7 Project Aim

***“The aim of this research project is to design & develop a title generating framework which attract users”.***

This project will provide a solution for writing headline to reach audience. By achieving this goal audience user engagement will be increased in different aspects.

## 1.8 Motivation

Being an online news reader whose reads article, blogs and news in different areas, has come across problem that common title writing practice using creative words and sentences also there isn't a framework to automate process of generating creative titles. Since I've known the importance of creative sentences in marketing and advertising decided to research this area and develop a framework.

## 1.9 Objectives

### 1.9.1 Research Objectives

#	Objectives
1	<p><b>Study about creating creative titles.</b></p> <ul style="list-style-type: none"> <li>• Research how popular content writers come with great title.</li> <li>• Study different types of title and its impact on audience.</li> </ul>
2	<p><b>Research about title with different creative text.</b></p> <ul style="list-style-type: none"> <li>• Examine the impact of same news with and without creative text.</li> <li>• Study how to embed creative text in title.</li> </ul>
3	<p><b>Research about best way identify different types of creative texts</b></p> <ul style="list-style-type: none"> <li>• Study about different types of creative text</li> <li>• Find a best way to identify wide amount creative text types</li> </ul>

<b>4</b>	<b>Test identifying creative text identifier</b> <ul style="list-style-type: none"> <li>• Test the accuracy of identification level.</li> <li>• Test the performance of identification level.</li> </ul>
<b>5</b>	<b>Examine suitable data sources to be used</b> <ul style="list-style-type: none"> <li>• Compare different data sources and choose suitable data source considering project scope, duration and related context.</li> </ul>
<b>6</b>	<b>Research title generation from contents.</b> <ul style="list-style-type: none"> <li>• Study about different title generation approaches.</li> <li>• Find suitable one based on project requirements.</li> </ul>
<b>7</b>	<b>Study about identifying keywords in title.</b> <ul style="list-style-type: none"> <li>• Study titles identify what type of words are added to attract audience.</li> <li>• Come up with a way to identify the keywords.</li> </ul>
<b>8</b>	<b>Research about embedding selected creative text in title without changing title's context.</b> <ul style="list-style-type: none"> <li>• Research different NLP approaches for adding and changing words in title without changing the context.</li> </ul>

*Table 1.1 - Research objectives*

## 1.9.2 Personal Objectives

#	Objectives
<b>1</b>	To improve time management skills
<b>2</b>	Improve document writing skills.
<b>3</b>	Improve problem solving skills.
<b>4</b>	Learn how to apply software development methodologies
<b>5</b>	Improve critical thinking & research knowledge.
<b>6</b>	Improve Technical and Coding Skills
<b>7</b>	Learn Software Development Principles, Architectural Styles and Best Practices in Coding

*Table 1.2 - Personal objectives*

## 1.10 Project Scope

### In Scope

- Derive sentences from corpus
- Identify important keywords to title.
- Identify creative text in data source.
- Identify relevant templates for title.
- Generate creative title for given document.
- System only works for sentences in the English language.

### Out Scope

- Not identify grammar mistakes.
- Not identify slang words.
- Not identify other than English language.

## 1.11 Resource Requirements

### Software requirement

- Microsoft Windows 7+
- Pycharm IDE
- NLP libraries & tools
- Microsoft Office Package
- Mendeley (Reference Manager)

### Hardware requirement

- Core i5 or above processer
- Minimum 8GB RAM
- 10 GB storage capacity

## 1.12 Rich Picture of the project

For better understanding about the framework rich picture illustrated with end application point of view.

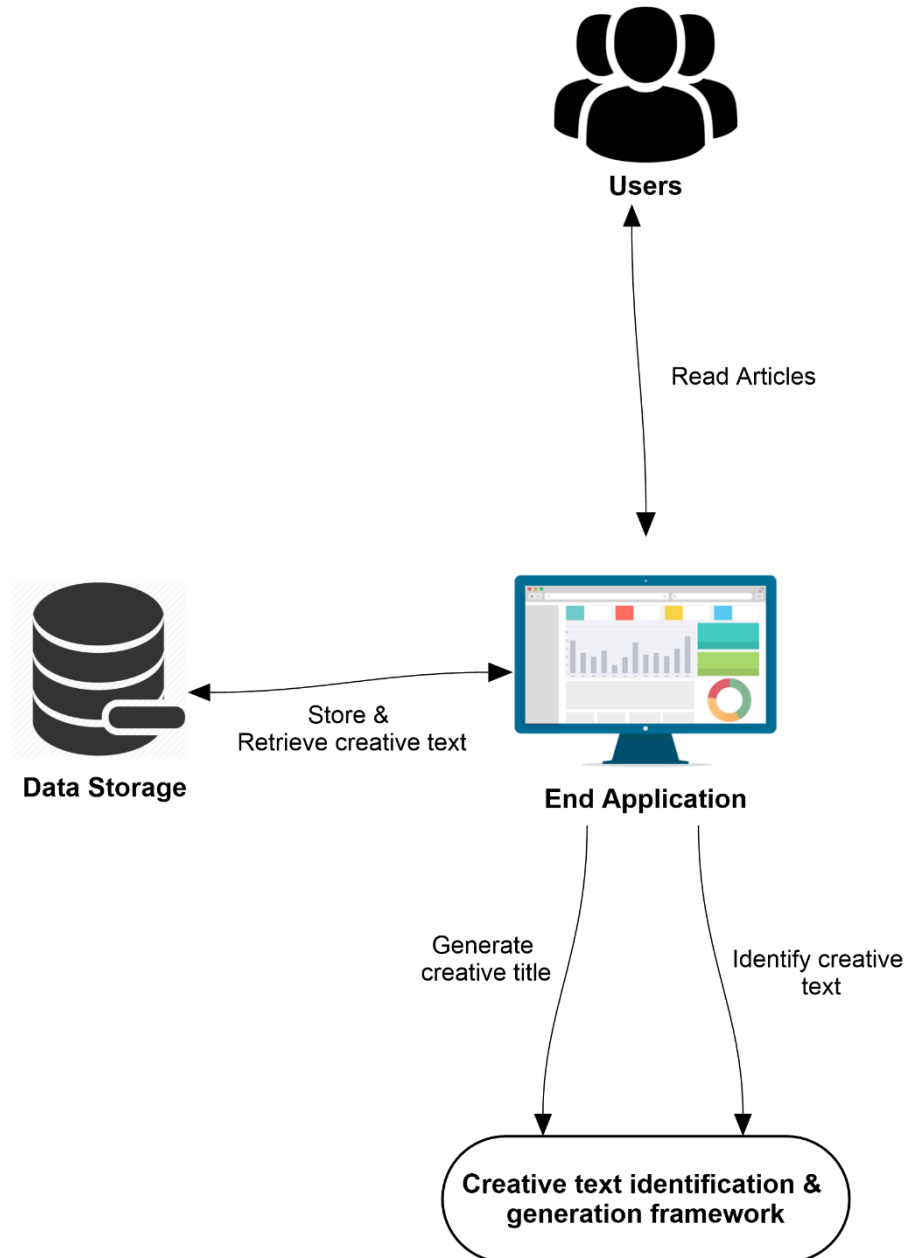


Figure 1.2 - Rich Picture

## 1.13 Chapter Summary

This chapter presented detail description about the research and project it's insight. Chapter initially describe about project background on importance of the information overload and readers attraction towards information for content writers and journalists. Then problem domain discussed about different tool and system available for creating articles, news, books and documents and importance of title generation among them. Moreover, project aim, scope and objectives defined to show the project boundary and requirements. Finally, rich picture provided to see the project in bird eye view.



## **Chapter 2: Project management**

## 2. Project Management

### 2.1 Chapter Overview

This chapter discusses about project management process of the project. Chapter begin with analyzing data gathering approaches of the project then continues to discuss about selecting suitable project management methodology. Then it continues to discuss about time allocation and constraints of the project. Finally, chapter discuss about choosing suitable research and development methodology for project by comparing different approaches.

### 2.2 Data Gathering Approaches

The two main data gathering approaches are,

- Primary Data collection – Observation, Interview
- Secondary Data collection - Web Crawling, Existing Data (Data warehouse)

Identified major data collection methods for review are observation, web crawling and existing data. Data collected from observation is more trusted because it's been observed from known area. Problem this method is it requires more time to collect data and change data into usable for project since the project time period is limited it is not feasible to use this method. Web crawling is another way collecting data. Web crawling data is not secure because different kind of data are collected it may affect the outcome of the project also it takes more time as observation so it also not suitable. Existing data is suitable data collecting method because it doesn't take time, data can be trusted because it used by several people and data cleaning also easy. Based on the review existing data is identified as suitable data collection method.

### 2.3 Project Management Methodology

Any project would have a scope, duration and cost as constraints. To properly manage these a suitable project management is required. In this project, requirements change frequently which highlights the importance of the choosing suitable project management. Considering project requirement following project management methodologies evaluated.

- PRINCE2
- PMBOK
- Agile

Considering project scalability, quality of final product, demand on learning about methodologies, development methodology and decision making, PRINCE2 methodology is identified as most suitable for the project.

### 2.3.1 Time allocation

Following table represents an overview of the identified main tasks of the project (refer appendix for complete Gantt chart).

#	Task Name	Duration (days)	Start	End
1	Preparation of Project Initiation Document	77	20-08-18	04-11-18
2	Literature review	60	05-11-18	03-01-19
3	Requirement Specification	18	04-01-19	21-01-19
4	Design	15	22-01-19	05-02-19
5	Implementation & Testing	50	23-01-19	14-03-19
6	Evaluation	14	15-03-19	28-03-19
5	Project Closure	20	29-03-19	17-04-19

Table 2.1 - Tasks of the project and time allocation

### 2.3.2 Constrains

Constraints of the project are reflected in our estimations. As long as the project operates inside the agreed constraints considered project on target. Good project management practice requires to define ranges for constraints. Following constraints identified for the project.

- Time availability – Project has to deliver in limited time, this can directly affect the research and implementation of the project. So, this considers as major challenge.
- Lack of prior knowledge – lack of knowledge in NLP area and automating process of title creation may affect quality of final outcome.
- Resource availability – Due to lack of resource available in NLP area and limited access to other resources can direct impact with implementation stage.

## 2.4 Potential Risks and Mitigation Plan

<b>Risk No. 1</b>	Available time for research and implementation of project is limited hence, identification failure of core functionalities of the project may affect other phases of project and final outcome quality.		
<b>Risk Level</b>	High	<b>Probability</b>	High

<b>Mitigation</b>	Identify the core functionalities in requirement gathering stage. Develop the core functionalities first since they are the fundamental features of project. If there is enough time develop other functionalities otherwise included as future enhancement.		
<b>Risk No. 2</b>	Lack of libraries and resources availability to implement the functionalities. This will interrupt process implementation.		
<b>Risk Level</b>	High	<b>Probability</b>	Medium
<b>Mitigation</b>	Get help from domain experts. Research about the available libraries and resources and adjust the implementation place based on availability.		
<b>Risk No. 3</b>	During the project time there can be chances where advance technology or existing system produce more accurate or reliability results.		
<b>Risk Level</b>	Medium	<b>Probability</b>	Medium
<b>Mitigation</b>	Be update about latest and current researches in title generation area. Frequently check technologies related to domain area. Follow domain experts and past researches and check latest development.		
<b>Risk No. 4</b>	Developing invalid features for proposed solution.		
<b>Risk Level</b>	High	<b>Probability</b>	Medium
<b>Mitigation</b>	Conduct user surveys and interview with experts (marketing experts, content creators) and make sure features are valid and usable.		
<b>Risk No. 5</b>	During the project time there can be chances where advance technology or existing system produce more accurate or reliability results.		
<b>Risk Level</b>	Medium	<b>Probability</b>	Medium
<b>Mitigation</b>	Be update about latest and current researches in title generation area. Frequently check technologies related to domain area. Follow domain experts and past researches and check latest development.		
<b>Risk No. 6</b>	Proposed framework doesn't match standard of requirements. A framework acts as tool to supply structure and template for constructing an application. If proposed framework doesn't provide the standards it won't be a valid implementation.		

<b>Risk Level</b>	Medium	<b>Probability</b>	Medium
<b>Mitigation</b>	Before begin implementation study about framework structures and check existing framework feature and implementation.		
<b>Risk No. 7</b>	Changes in requirement in different phases of project. It may difficult to manage all requirement changes with available time.		
<b>Risk Level</b>	High	<b>Probability</b>	High
<b>Mitigation</b>	Make flexible research and implementation plans to make able to change in time. In case of limited time apply only critical changes.		

*Table 2.2 - Risks and Constraints of System with mitigation plan*

## 2.5 Research Methodology

Inductive and Deductive are the broad methods of research. Concerning both methods inductive method is most suitable for this project when compared to deductive. A deductive method begins with hypothesis and it emphasize causality but inductive aim to narrow down the scope of study and it focus on finding new areas or looking previous works in different viewpoints.

## 2.6 Development Methodology

Before starting a project, a proper development methodology must be chosen. Maintaining a development methodology helps to structure the project properly, deciding the phases on duration, make decision and develop system according to plan. To select the most suitable development methodology for system, characteristics of followed development methodologies were evaluated against project's requirement.

- Waterfall methodology
- Rapid Application methodology
- Spiral methodology
- Agile methodology

Waterfall methodology is linear module hence, the requirement of the project must be stable and cannot be changed once project initiated. Research project requirements change by nature. In this project requirements change in phases i.e. literature review, development decisions and features. Therefore, traditional waterfall method is not suitable it requires more methodology which has more iterative approach. Considering Rapid Application methodology, it has minimal planning in its initial stage and quickly moves in development of prototype in its early stages. This method requires certain level of knowledge about final product to develop

prototype. The project requires more planning and it doesn't have much knowledge about final product in early stages throughout the time it understands about it hence, Rapid Application method is not suitable for the project. Agile methodology includes more iterative approach and incremental development but it is mostly suitable for huge projects where it has big vision and several stockholders. Also, agile method requires lot of time availability because of sprints, customer interaction and discussions. Since this project has fixed and limited time and scope agile method is not suitable. On other hand Spiral method capable of changing requirements in changes. It is a combination of iterative and waterfall methods with significant accent on risk analysis. In spiral development stage scalable to make changes in functionality. As major challenge of this project is time availability and requirement change spiral method is most suitable for development.

## 2.7 Design Methodology

Structured and Object-oriented are the main analysis and design methodologies. Both methods reviewed to identify suitable analysis & design methodology for project. Structured analysis and design both are based on functionality. Process of structured method begins with identifying overall purpose then functional decomposition is done for developing. Benefits of this method is it follows top-down approach which is easier to understand. Since, it's based on functionality it gives better understanding of the system. Drawback of this method is it doesn't support code reusability, so the cost and time of the system is high. Also, update and maintenance of the project would be exceptionally costly because of measured quality and institutionalization isn't usable.

On other hand Object-oriented method focuses on data rather than procedures. This method has several benefits. This method concern about code reusability so the time and cost of software development would be less. Also, it allows effective management of software complexity. Drawback of this method is it's restricted within objects which may cause problem for systems which are intrinsically procedural or computational in nature.

Based on the comparison Object-oriented method identified as suitable design and analysis methodology because of changing requirements of project, cost, time and past experience.

## 2.8 Evaluation Methodology

Both quantitative and qualitative evaluation methods will be used in this project to provide the best overview of the project. For the project's qualitative evaluation, feedbacks will be collected for different emulation goals from domain experts and technical experts via surveys and questionnaires. Whereas the quantitative evaluations are done to measure the accuracy from

test results, techniques, statistical evaluation and survey results from domain experts and technical experts.

## **2.9 Work Breakdown Structure**

Attached in [Appendix A](#)

## **2.10 Activity Schedule**

Attached in [Appendix K](#)

## **2.11 Chapter Summary**

This chapter represent the project planning and selection of different methods. The chapter initially discussed about data gathering approaches and decided to use existing data with comparing other methods. Then project management methodology defined, and PRINCE 2 method was chosen. Time constraints and risks and mitigation plan provided for the planning. Selection of research, development and design methodologies also discussed with comparing different methods with advantages and disadvantages. Finally, WBS and Activity schedule provided for better understanding on planning of the project.

## **Chapter 3: Literature Review**



## **3. Literature Review**

### **3.1 Chapter overview**

The chapter describes the creative title generation in depth. The proposed approach decomposed into three tasks: identify creative text, generate title and generate creative title from given document. Chapter begins with describing fundamental idea about automatic title generation and continues to discuss about classifications of title. Then it critically analyzed related studies about title generation, creative text generation and creative text identification while discussing approaches affects over user encouragement provided to select the most appropriate approach. Afterwards, approaches, techniques and theories of division of framework analyzed and criticized to identify appropriate approach to implement. Presented topics were compared and contrasted with advantages and disadvantages.

For simplicity, article, blog, magazines or news, etc... mentioned as document in the rest of the review.

### **3.2 Automatic title generation system**

Creating title for a document is an art. Journalist, writers and editors review and rewrite title for document multiple times to come up with great title. Title generation system automate this process to help them to effectively do their job. Goal of title generation system is to produce a title that grab attention of the readers.

Title generation system faces several challenges, in natural language processing (NLP) certain linguistic aspects like unusual use of tenses and deliberate ambiguity, Title are usually short hence; NLP tools rely on producing short text is less and also studies on title generation is less (Piotrkowicz, 2017).

### **3.3 Classification of title types**

Titles can be classified into three types indicative, informative & attractive (eye-catchers). A title that indicates what topics covered by the document called indicative titles, informative titles are titles that reflect the main story, event or theme covered by the document and attractive titles which are primarily focused to attract and encourage people to read the document (Gattani, 2007). Title generation systems use title type one over other another type based on their purpose and target audience.

Informative title type is mostly suitable for a situation where readers need to know the document's content and purpose to reflect in the title. More like research journals and articles. Most of the time content creators pick attractive title type rather than indicative and informative

because of it used to increase the user engagement and attention towards the document. Vast amount of contents available on online and the competition among content creators to get audience to read their document is high. This is the one of huge challenge faced by content creators. They mostly use attractive title type to overcome the competition (Piotrkowicz, 2017). A research done in eye tracking in online news found that many people “entry-point reader” are viewed article based on article’s title with minimal reading time (Pan *et al.*, 2007) informative and indicative title type won’t be used to get people to read document but attractive title does. The attractiveness can make user read document.

On other hand attractive titles also come with some drawbacks. Content creators follow some strategies to lure audience to pick their document, in particular Clickbait “titles which are sensationalized, turn out to be adverts or are simply misleading” (Frampton, 2015) and “forward-referencing”: title that contain messages as surprise, gossip like, super natural and sensationalism. When content creators use these in their document’s title, they may lose followers and trust in the brand value and readers may mislead to documents which is not expected or they fall in trap that might lead to click-jacking attacks (Blom and Hansen, 2015; Shu *et al.*, 2015). Attractive title type comes with strengths and weaknesses compared to other types. Situations where user traffic is high attractive title type is mostly suitable.

## **3.4 Related Studies**

### **3.4.1 Title generation system related studies**

Most previous studies on title generation can be divided into rule-based, statistical and summarization-based approaches. Approaches are compared along with the pros and cons of each category based on problem domain.

#### **Rule based approaches**

Rule based approaches produce title using handcrafted linguistically-based rules by detecting or compressing important content of document (Colmenares *et al.*, 2015). It doesn’t require training data to generate title because there isn’t a model to be learned. A main purpose of using rule based approach is that it produces syntactically correct title (Shao and Wang, 2017). Rule based approach chooses a central sentence for title that represent the main theme or event in document. Which is not a good practice because important pieces (event, story or theme) spread throughout the article and trimming or detecting single important sentence might lead to miss important information. Rule based approaches lacks in exploring complex relationship in the sentences (Gattani, 2007; Colmenares *et al.*, 2015). Rule-based methods able to generate all types of title because of set of rules can make flexible changes generation.

Shao and Wang (2017) implemented a title generation system DTATG (Dependency-Tree Automatic Title Generator) using dependency tree compression model. DTATG produce title in two phases. First, it finds the central sentence of the document using RAKE keyword extraction method and Second, the central sentence gets assigned to dependency parser. It compresses the sentence with set of empirical rules. DTATG works in open domain area, it doesn't rely on training data and less complex compared to statistical systems. It doesn't generate attractive title in every situation but when it does, the title contains words that appears source document. based on previous statement generated title attractiveness is depend on words in source document.

### **Statistical based approaches**

As noted by Gattani (2007) statistical or learning approaches are built from large annotated training corpus (title-document pairs) and works with supervised learning. The model learns the relation between title and document content and the learned model is applied to create title for new documents.

This approach overcomes the limitations problem with rule-bases and summarization approaches. Statistical based title generation can generate text which are not in source document (Alfonseca, Pighin and Garrido, 2013) when compared to rule based approach this is an important advantage. This is heavily relying on training data unlike rule based and summarization hence, the accuracy of the title depends on how much effectively the training corpus annotated. On other hand, learning from training data can be an advantage compared to text summarization because sentence selection rules are more or less hand crafted and the system learn by itself (Jin, 2003). Statistical approaches are domain specific. When the model learned from specific domain it's generates title based on domain which cannot satisfy other models sometimes for an example, news related document's title usually reflects the main or interesting event when a statistical model learns from this domain and generates titles for a different domain blog it generates title same way as news domain which is not suitable for blog domain. Statistical approach learns the correction between the title and every word in each document so, it is more expensive and requires rich computer resource than other approaches (Gattani, 2007).

As per Gattani's (2007) opinion, most of statistical based approaches generate indicative title type because statistical approaches read the entire words in document and choose words for title.

Machine learning approaches used to identify the drop words on trimming process Unno et al., (2006) applied maximum entropy model to learn on context-free grammar (CFG) parse

tree method. The compression part removes additional and optional words and identifies short sentence. This process guarantees the sentence is grammatical by assigning probability values. Maximum entropy model is the extension of Knight and Marcu's noisy channel model (Knight and Marcu, 2000). This method can produce accurate compressing sentence since it is learning. Also, it resolves the issue with compressing negative words using maximum entropy model.

### **Text summarization approaches**

Summarization approaches summarize document with short length based on number words or ration between documents and title then define it as title (Gattani, 2007; Ayana *et al.*, 2017). Human summarize document using their language knowledge and past knowledge in summarizing but for the computers it's a challenging task because it doesn't have the knowledge of human and language capability hence, it's a difficult task to build a summarization system (Mehdi *et al.*, 2018). Text summarization approach can work in open domain area. The main advantage of text summarization is its simplicity. We can get an existing text summarization method and ask to compress document with high compression ratio and use produced summaries as title (Jin, 2003). When text summarization ration goes below 10% the quality title of summarization decreases. Usually a title of document contains nearly 10 words for document of more than hundred words which means text summarization title sometimes generates low quality title (Jin, 2003). Most modern summarization systems generates title by just recycling and reordering the words in source which can raise a risk because it might change contextual meaning of document produce different outcome (Colmenares *et al.*, 2015) which are main drawback of text summarization.

Text summarization methods generates title more natural to editor's work in natural language. The summarization of a document contains the main event or theme in the title which very beneficial because it catches the attention of the readers. Jin explained that text summarization method produce title with the main event and theme of the document, and it has to catch the attention of the readers by providing this example. News story about the results of an NBA regular game that the Wizard beat the Net and short text summary would be "The Wizard beats the Net with 108 to 97" and how an author might produce title as "Jordan flies again" which doesn't explain about the main event about the game. He argues that text summarization title is better than author's title (Jin, 2003).

Considering approaches of title generation, rule-based approaches have advantage of easy implementation, required less resource and not depend on dataset but they have drawback that is the accuracy of capturing the main event or theme in document is less compared to other approaches hence rules-based approaches aren't suitable. Statistical and

summarization approaches both produce accurate title but statistical approaches specific to a domain and its accuracy is depend on trained data set. Summarization approaches overcome these drawbacks. Considering these facts summarization approach is most suitable.

### **3.4.2 Creative text identification related studies**

The creation of creative sentence or phrase are often based on concept of blending theory (Fauconnier and Turner, 2001). It fundamentally a method of evoking a secondary concept (which explains the situation in different perspective while keeping the original expression) to replace the primary concept. Fields such as advertisement, news production and poetry make use of this in their area to reach many audiences. The creative sentences can be in many forms such as metaphors, idioms, similes, expressions and more. Related works mostly related to specific domain in creative sentences.

#### **Dictionary approach**

Dictionary approach is based on word's meaning and relation. Creative sentence identified how words related in different relationship and their frequencies (Muzny and Zettlemyer, 2012; Verma and Vuppuluri, 2015).

Verma and Vuppuluri (2015) introduced a novel approach to identify idioms on text based on dictionary and web knowledge. This system first extract phrases from sentence then phrases redefined by gather information from dictionary and web finally idioms identified via performing statistical methods subtraction and union on phrases. The method is unsupervised and not biased to a specific domain and this may lack when information gathered from dictionary and web is not related. This might affect the accuracy of the system.

#### **Selectional Preference approach**

Conceptual blending in a sentence can be effectively identified through selectional preference method. Selectional preference is tendency of words to co-occur with preferred words from certain semantic category. Certain verbs in sentence constraint to their arguments that they take, conceptual blending in a sentence can be exposed using violation of selectional preference (unless it's a grammatical mistakes).

Selectional preference has been a popular research subject in the NLP community. As Light and Greiff (2002) say "Words in the same sentence stand in relationships with one another". Using selectional preference violation, researches identified metaphors, metonymies, anomalies, idioms and expressions in a sentence.

Example of selectional preference violation,

(1) “*My aunt always drinks her tea on the terrace*”.

(2) “*My car drinks gasoline*”. (Wilks, 1978)

In above sentences the verb “*drinks*” prefers a subject type “*animate*” and grammatical object type “*liquid*”. The sentence (1) satisfies the constraints but (2) violations the constraints of selectional preference, “*drinks*” taking car is a figurative sentence and indicates a creative sentence usage (Shutova, Teufel and Korhonen, 2013).

Haagsma and Bjerva (2016) used selectional preference violation to identify metaphors in text. The method is based word occurrence frequency. They use clustering method on words to increase the coverage of metaphor identification. Metaphor in sentence identify by how often words in sentence co-occur. Since the method uses clustering accuracy of identification increases. The only drawback of the system is it doesn't consider semantic similarity of words.

When comparing the approaches of creative sentence identification, machine learning approaches' accuracy is extremely depending on the dataset and most of the time it identifies sentences belong to specific domain. On other selectional preference and dictionary method overcomes machine learning approaches drawbacks. Selectional preference approaches has a main advantage over dictionary approach that its different types of creative sentences (metaphors, metonymies, anomaly and other) it means it over generates with respect to a type of creative sentence (Shutova, Sun and Korhonen, 2010; Shutova, Teufel and Korhonen, 2013). Based on the arguments it is identified that to identify creative sentences in text selectional preference method is more suitable.

### 3.4.3 Creative sentence generation related studies

Creative title generation related to different areas such as poetry (Toivanen *et al.*, 2012), story, slogan (Gatti *et al.*, 2015) and humor generation (Valitutti, Stock and Strapparava, 2009) domains.

Creating creative sentence for given specification which specific to a domain is a challenging task. Mungala et al. (2018) proposed a framework PersuAIDE!. PersuAIDE! produce various forms persuasive sentence for given list of keys with pre-attached persuasive key-phrases. Using a neural language model to identify matching persuasive key-phrase to keywords then it replaces the text with persuasive key-phrases. PersuAIDE! is specific to a domain and users required add persuasive key-phrases by themselves. It heavily relies on different data corpus.

BRAINSUP is an extensible framework for generating advertise and catchy sentences developed by (Özbal, Pighin and Strapparava, 2013). It generates slogans for given phonetic target words and user can manipulate desired slogan by forcing some values to show in

sentence such as rhymes, alliteration and plosives with combination of these properties it generates sentence. It deeply relies on syntactic information such as words and templates to generate well-formed sentence. Within generation process of BRAINSUP, it selects the most suitable template for matching input keyword, and it fills the empty places in the template from related words from input.

Lexical substitution is another widely used method for sentence generation. Gatti et al. (2015) developed a lexical substitution based slogan generation framework Slogans are not forever. The framework retrieves recent news title and matching well know expression from database generate a slogan by substituting words. The key component of behind this framework is identifying suitable words for substituting. Using dependency statistics, it identifies suitable words using relation statistics among the words.

Most reaches in creative systems are based on conceptual blending theory. It means evoking a secondary concept with source text to form new text, which inherits properties from starting domain (Fauconnier and Turner, 2008).

Existing systems generate sentences with the help of predefined and well-known properties such as templates, key-phrases and expressions and these properties are not available to add, change or update which is drawback of the systems. Being able to identify creative text from documents makes framework more beneficial. Without adding expression explicitly to system, use a corpus and let the system do the identification and storing expression would be a time saving and flexible feature.

### **3.5 Creative sentence identification approaches**

As discussed in [Section 3.4.2](#) selectional preference approach effectively identify creative sentence in text. Selectional preference is a corpus-based method. Prior work on selectional preference was conducted by (Resnik, 1996). The research is depending on training data and a class hierarchy such as Wordnet. This model successfully applied to number of NLP task including word sense disambiguation.

Number of researchers used Resnik's selectional preference approach. Metaphor identification systems (Shutova, Sun and Korhonen, 2010; Haagsma and Bjerva, 2016) uses selectional preference violation to identify metaphor in sentence. The research follow the idea of (Wilks, 1978) that metaphor contains a violation of selectional preference in a given context.

In selectional preference method, it is not possible to identify full coverage of SPS between word pairs in large corpus. This could cause problem when computing SPS for an infrequent word combination. Solution to this problem is to do some sort of generalization over the words.

Generalization approach can be divided in three divisions WordNet-based, clustering based and neural networks-based approaches.

### **WordNet-based Approach**

This method is depending on set of training data and a class hierarchy such as Wordnet. The system computes distribution of argument semantic classes of a predicate using WordNet. For each argument system collects candidate semantic classes suitable for predicate and picks semantic classes from the candidate with maximum selectional association score.

WordNet approach outcome is more similar to human judgment. Also, implementation is easier compared distributional approaches. WordNet approach has drawbacks that this model generalization method entirely relies on WordNet hence, it causes number of problems. As Rooth describes “entailment hierarchies are presently available for few languages, and we regard it as an open question whether and to what degree existing designs for lexical hierarchies are appropriate for representing lexical meaning” (Rooth *et al.*, 1999).

### **Clustering based Approach**

Prior work on clustering approach is done by Rooth et al. (1999). He proposed Expectation-Maximization (EM) based clustering method for selectional preference. Recent researches in selectional preferences, researches used different kind on clustering methods spectral clustering (SPEC) (Shutova, Sun and Korhonen, 2010) k-means, Brown clustering (Haagsma and Bjerva, 2016).

K-means clustering is the most well-known clustering algorithm in recent years. It is unsupervised and fast. Required classes or groups has to be given before process begin. It has few computations in clustering process that it computes the distances between points and group centers. The complexity of k-means is linear. K-means also has disadvantages that it begins with random clusters that means sometimes the class or group may end up in different cluster results on different runs of the algorithm. And required classes or groups has to be given at beginning this may cause different results usually cluster algorithm is responsible for finding clusters amount.

Brown clustering (Brown *et al.*, 1992) is another clustering method for NLP. It groups similar words using distributional information. Brown clustering is unsupervised, it induces a hierarchical clustering over words to form a binary tree. This implementation has  $O(v * m^2 + n)$  complexity. Where corpus contains  $n$  words and  $v$  unique words and the  $m$  is the fixed window size. Brown cluster's accuracy depend on the training corpus.



## Neural Network Approach

Recent research in selection preference are mostly related to neural network. Neural network learns the relation between words and identifies selection preference. This approach requires training corpus. So, the accuracy is depended on the dataset.

Van de Cruys (2015) developed a neural network which learns to separate between felicitous and infelicitous arguments for a particular predicate. The model is unsupervised and it learns from unannotated corpus. They proposed two neural network architectures one that handles standard two-way selectional preferences and one that is able to deal with multi-way selectional preferences.

Neural network approach has accurate result of selectional preference, but it trained on corpus hence the accuracy depend on trained corpus domain. Clustering approach has advantage of that this method doesn't require a fixed taxonomy of semantic relations (WordNet) and unlike WordNet based approaches it works with any language. The problem with this method is the words classes collected through clustering doesn't always semantically similar, sometimes no noticeable relationship with class members also be there. As Resnik explains "It would seem that the information captured using these techniques is not precisely syntactic nor purely semantic — in some sense the only word that appears to fit is distributional" (Resnik, 1993).

## Selectional Preference Measure

Most selectional preference approaches are based on measure proposed by (Resnik, 1996). It uses unsupervised clusters to generalize over seen arguments. Selectional preference strength (SPS) defined as the difference between the prior distribution and posterior distribution. SPS indicates how the selective verb is in the choice of its arguments. It calculated as Kullback-Leiler (KL) divergence.

$$SPS(v) = D(P(c|v)||P(c)) = \sum_c P(c|v) \log \frac{P(c|v)}{P(c)}$$

*Equation 3.1 - SPS calculation*

$P(c)$  is prior probability of noun class and  $P(c|v)$  is the posterior probability of the noun class. The difference between  $P(c)$  and  $P(c|v)$  use relative entropy (or Kullback-Leibler distance). It described as amount information that a predicate tells us about the semantic class of its arguments. SPS method is more useful when it used with selectional association (SA) method. It identifies how an argument is co-occur with a predicate.

Selection Association defines as:

$$SA(v, n) = \frac{1}{SPS(v)} * P(n|v) * \log \frac{P(n|v)}{P(n)}$$

*Equation 3.2 - SA calculation*

Selectional association quantified as the relative contribution of the class towards the overall selectional strength of the predicate.

## 3.6 Creative title generation approaches

Lexical substitution identified as suitable method for creative title generation task. This method became popular for evaluating lexical inference models since SemEval-2007 (McCarthy and Navigli, 2007). Purpose of this method is to predict substitute words for target words without changing the meaning of the context. As an initial phase semantic words for target retrieved as candidates then it identifies words for context using different similarity measures.

### 3.6.1 Semantic words

Word embeddings and wordnet methods are popular methods for finding semantic words for lexical substitution task.

#### Word Embeddings

Word embeddings are vector representations of word types. Popular word embedding models are Word2Vec (Mikolov *et al.*, 2013) and Glove (Pennington, Socher and Manning, 2014). Both models learn vectors from co-occurrence information. But they differ in that Word2Vec model is a predicative model and Glove is count based model. In Word2Vec mapping between the target word to its context word implicitly embeds the sub-linear relationship into the vector space of words, so that relationships like “king:man as queen:woman” can be inferred by word vectors. Glove adds some more practical meaning into word vectors by considering the relationships between word pair and word pair rather than word and word. Glove enforce the word vectors to capture sub-linear relationships in the vector space. It proves to perform better than Word2vec in the word analogy tasks. Number of researchers used word embeddings as prior words for task (Melamud, Levy and Dagan, 2015).

#### WordNet

WordNet (Fellbaum, 2010) is a lexical database for English language. Using synonyms, hypernyms and hyponyms features researches retrieved candidate words for substitute task (Kremer *et al.*, 2015).

### 3.6.2 Similarity measures

Similarity measure is the measure of how much alike two text objects are. Euclidean and Cosine are the popular among similarity measures.

#### Euclidean distance

Euclidean Distance is only calculated over non-NULL dimensions

#### Cosine similarity

Cosine similarity calculates similarity by measuring the cosine of angle between two vectors. Metric of cosine finds the normalized dot product of the two attributes. Cosine similarity generally used as a metric for measuring distance when the magnitude of the vectors is not important. This usually happens with text data represented by word counts.

## 3.7 Title generation approaches

As discussed in [Section 3.4.1](#) summarization method identified as suitable title generation. Text summarization approaches can be broken into different categories based on purpose.

- Single document vs multiple document – provide summary only for single document or else summary several documents.
- Generic vs Query-focused – provide summary without assuming anything about domain or treat summarize contents and all input equal and summary contains information only given input text.
- Extract vs Abstract - summarization approach represent sentence and words from original document vs approach are based understand the original document and produce summary.

### 3.7.1 Abstractive Text Summarization

Abstractive summarization represents the actual work done in natural language. It generates title with words that are not appeared in source document and it generates title with short length and conveys most critical information in source document.

Abstractive summarization generates title entirely new way, but it is not reached the maturity of extractive summarization (Radev and Erkan, 2004). It comes with some drawbacks such as semantic representation, inference and natural language generation which are relatively harder than data driven approaches such as sentence extraction. The approach is not completely abstractive, as Mehdi et al. (2018) say “existing abstractive summarizers often rely on an extractive preprocessing component to produce the abstract of the text”.

HEADS is a title generation system developed by (Colmenares *et al.*, 2015). Its main focus is to generate title in way of editors summarize document in real world. HEADS is developed from end-to-end encoder-decoder framework in neural network. It uses conditional random fields (CRF) sequence prediction method to train model. HEADS can generate more natural and short titles because the model learned to predict the next word in sentence. HEADS heavily rely on training data. Hence, the accuracy of title depends on training data. Problem with the system is it might generate title in correct sequence manner but with incorrect meaning which is a drawback of system.

Alfonseca, Pighin and Garrido (2013) developed HEADY. it identifies the events in document collection using pattern extraction and generates title. HEADY trained on Bayesian network on the co-occurrence of syntactic patterns. The main purpose of HEADY is to generate title in most compact, objective and informative way from given document collection. It identifies the same event in document collection and generates it hence, the title presents the event in informative way.

### **3.7.2 Extractive Summarization**

Extractive summary produces summary based on selecting relevant sentences from source document. The length of the title based on number of words or compression rate.

Extractive summarization is simple and robust method. It overcomes the problems of abstractive summarization. Drawback of extractive summarization is it generates summary with same words in source document which is not related compared to real work done by editors. Other than this issue extractive summarization produces accurate results (Radev and Erkan, 2004).

#### **Statistical based approaches**

Statistical approaches extract important sentences from document using different statistical features. Statistical features in sentences are title, location, term-frequency and more (Dhanya, 2013).

For each sentence, selected statistical features scores are computed and added together to get total score of each sentence. Then as per computed score sentences are extracted for summarization-based length or ratio of desired outcome.

The main advantage of statistical approach is its simplicity and easy to implement and it doesn't require complex linguistic processing or additional linguistic knowledge. Also, it requires less computer resource compared to other approaches. Statistical approaches aren't domain & language independent. Statistical approach isn't suitable to summarize professional

text documents such as health and medical because it may contains important words which appear few times hence, when summarizing statistical method may not filter out keywords (Beliga, 2014) which affects the accuracy of summarization other this issue statistical methods delivers intermediate results.

### Graph based approach

In graph based approach words or sentences are represented as nodes of graph and edges between nodes represent semantic similarity between them (Gambhir and Gupta, 2017). Once the graph constructed it finds the important sentence for summarization by finding sentences which are strongly connected to many other sentences, based on sentence ranking and desired length summary will be produced.

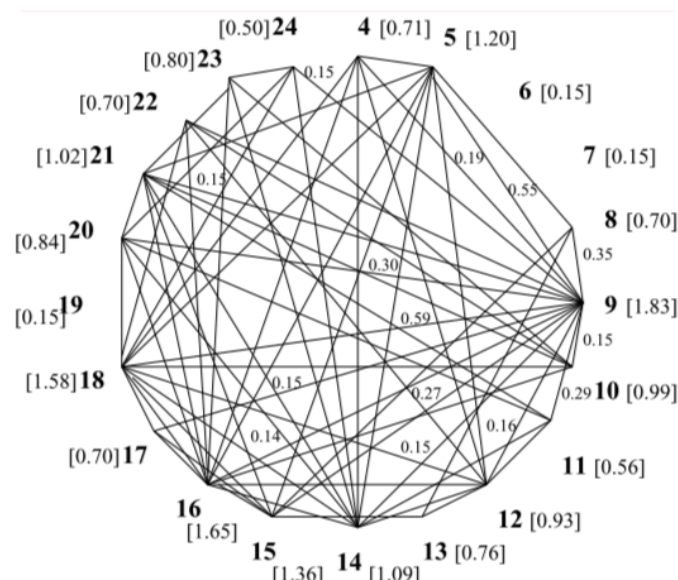


Figure 3.1 - Sample graph build for TextRank extractive summary as depicted in (Mihalcea and Tarau, 2004). Each node represents a sentence

Radev and Erkan (2004) developed a graph based summarization called LexRank. It used a connectivity matrix based on intra sentence cosine similarity between edges in graph. LexRank calculates the important sentences based on eigenvector centrality in a graph representation of sentences. LexRank perform well with multi documents summarization. Another popular graph based summarization is TextRank (Mihalcea and Tarau, 2004). TextRank identifies similarity between sentences based on common words among them. It mostly suitable for single document summarization. Both summarizations have common tasks. They are based on centroid-based summarization. TextRank and LexRank is derived from Google's PageRank (Brin and Page, 1998) a web page ranking algorithm. Both TextRank and LexRank select important sentences by performing random walk over the graph and weight each sentence. Summary formed by combining top ranking sentences.

Graph based summarization can be applied to any language because it doesn't required any language specific linguistic processing other than sentence and word boundary detection (Mihalcea and Tarau, 2005). Drawback of this method is it cares sentence similarity to generate graph not the understanding the relationship between sentences.

### **Topic based approaches**

This approach tell about events which occur frequently in document (Gambhir and Gupta, 2017). Topic based approaches works well when a topic information provided for summarization (Dhanya, 2013) otherwise it works same as other approaches without a purpose. This approach can work well with information retrieval systems because it provides a query (topic information) to retrieve information.

### **Discourse based approaches**

This approach makes use of linguistic features in text for summarization. It involves the analysis of semantic relation between text units. Research work in multi document summarizing, some researches involved in determining the important sentence by check cross-document. Radev and Erkan (2004) studied cross-document relations and introduced CST (cross-document structure theory). This model connects the words, sentences or phrases if they are semantically similar. It identifies relations such as Identity, equivalence, translation ext... between them.

This approach relies on linguistic analysis tools (a discourse parser, etc.) and resources (Word Net, Lexical Chain, Context Vector Space, etc.) (Dhanya, 2013). Since, it uses such high-quality tools and resource it requires high computer resource.

### **Machine learning approaches**

Machine learning approaches learn from training data (super vised or unsupervised) and apply the learned model to summarize document. Support Vector Machine (SVM), naïve Bayes classification, mathematical regression, decision tree and neural network are some supervised algorithms. Un supervised algorithm try to identify hidden structure in training data. Hence it suitable for newly observed data without complex modification. Clustering and Hidden Markov Model are some unsupervised algorithms (Gambhir and Gupta, 2017). This method requires complex and expensive computer resources. Since it uses training data it often dependent to domain and the model has to learn every time the domain changes (Beliga, 2014).

We discussed different extractive summarization approaches. Each of them has a purpose and goal and they are suitable for different situations based on their features. Machine learning based approaches summarize with knowledge from training data which has an additional

advantage compared other approaches because it generates summarization more natural to real writers, but the problem is that the accuracy of summarization depends on effectiveness of training data since there is no standard for selecting sentences for summarization people select various approaches to summarize. Also, machine learning approaches specific to domain which also a drawback of this approach. Topic base approach generates accurate summary when information of title is provided. It is not possible to give title information every time. Discourse based approach is more suitable for multi document summarization. Statistical and graph-based approaches have features of working on open domain and single document, but graph-based approach produces more accurate summary and it captures the main theme or event of document than statistical approach. Considering this information graph-based approach is more appropriate to generate title from given document.

### **3.8 Chapter Summary**

In this chapter, we find the suitable approaches, concepts and techniques of automatic creative title generation framework. Attractive title is identified as most suitable title type. Based related studies, approaches and problem domain suitable approaches selected - graph based summarization for title generation, selectional preference method for creative text identification and lexical substitution for creative title generation. At last it was resolved that none of the previous solutions have experimented the exact three approaches proposed in this project.

# **Chapter 4: Requirement Specification**



## 4. Requirement Specification

### 4.1 Chapter Overview

This chapter describes the system requirement specification process of the framework. The chapter begins with presenting the process of stakeholder identification and analysis in relation to the framework. Then analysis of different methods of requirement elicitation is discussed with executing suitable requirement elicitation methods and its outcomes. Finally, the chapter presents the scope refinement.

The chapter also includes the use-case diagram, use case descriptions, activity diagram, the identified functional and non-functional requirements

### 4.2 Stakeholders Analysis

Stakeholders of the system are identified to reduce the avoidance of implementation failure and analyze involvement, interest and impact of stakeholders in the system.

#### 4.2.1 Stakeholders Onion Model

Figure 1 depicts the stakeholders of the system and interactions among them. Interactions of among stakeholders are explained below.

- Supervisor gives advice to researcher in order to finish the project as planned standard.
- Developer community expects support and guide from project manager and expects features to be developed to meet their requirements.
- Project manager should concern about competitors to provide better service.
- Organization develops and maintains new application by applying framework (end product of this project) by via a developer (or organization).
- Organization earns revenue through developed application by providing service to customers.

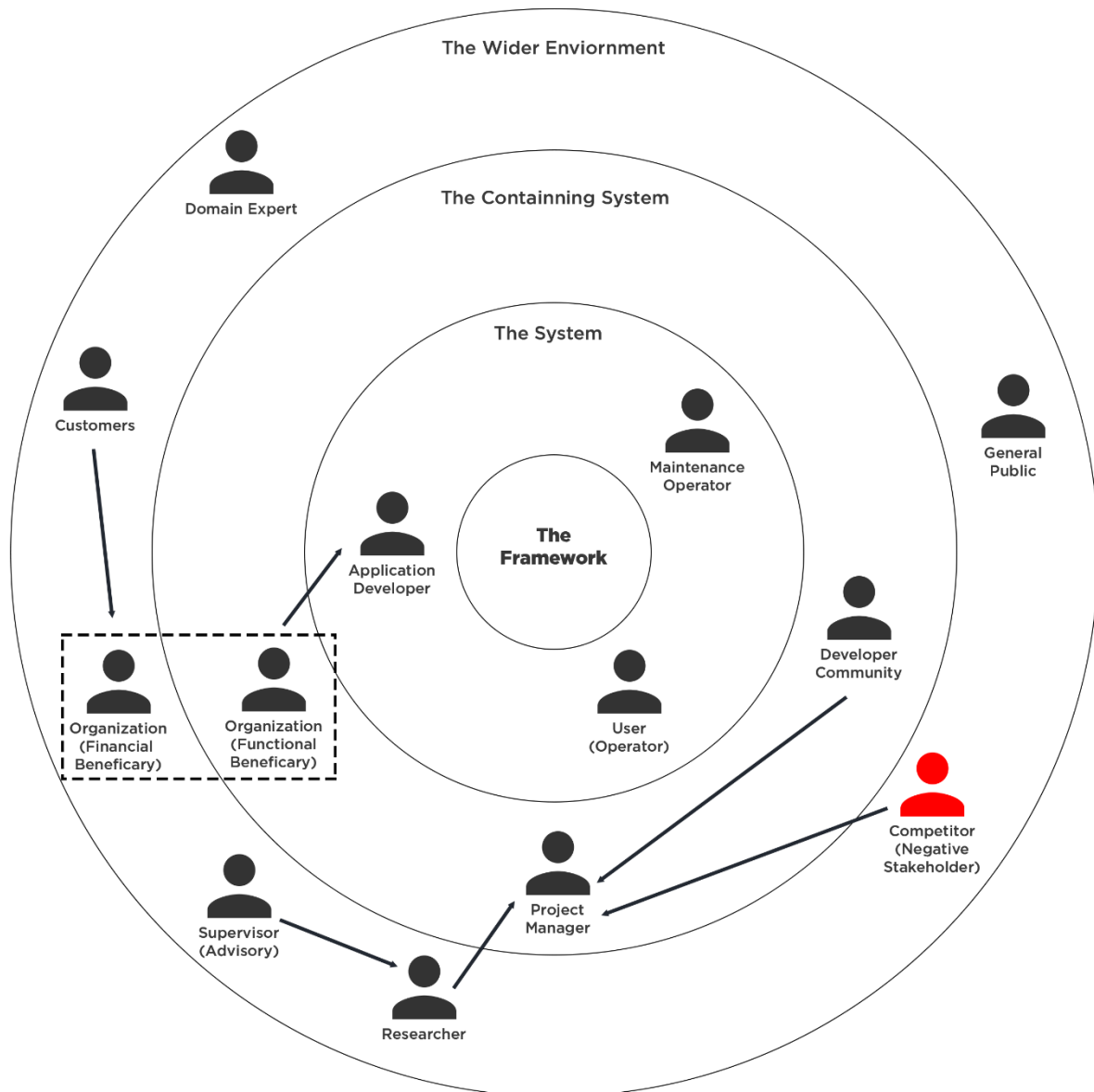


Figure 4.1 - Onion model

### 4.2.2 Stakeholders roles and description

Viewpoint and role of stakeholders discussed below as depicted in Figure 1

Stakeholder	Viewpoint
<b>Functional Beneficiary</b>	
<b>Developer Community</b>	Expects certain standards and features in product in order to meet requirements of their product. Expects assistance and guidelines to implement their product.
<b>Organization</b>	Appoint or hire application developer to implement product.

<b>User</b>	Gets benefits from end product.
<b>Maintenance Operator</b>	Feed corpus to system
<b>Customers</b>	Receive services provided by organization.
<b>Financial Beneficiary</b>	
<b>Organization</b>	Earn revenue by provide service with the assistance of product
<b>Researcher</b>	Expects revenue to from software purchasers.
<b>Purchase</b>	
<b>Organization</b>	Purchase framework to utilize work and apply framework (end product of this project) to existing or new system
<b>Managerial</b>	
<b>Project Manager</b>	Manages all constraints of the project to ensure appropriate flow
<b>Negative</b>	
<b>Competitors</b>	Wants to provide better service by identifying product and include additional or accurate features.
<b>General Public</b>	To point out strengths and weaknesses of the framework/application
<b>Regularity</b>	
<b>Supervisor</b>	Assist developer to finish project by providing advice and guidelines.
<b>Expert</b>	
<b>Domain Expert</b>	Provide expert opinion about the technologies and methodologies used for the project.

*Table 4.1 – Stakeholder roles and description*

### 4.3 Requirement Elicitation

Requirement elicitation process follow through to identify suitable requirements from stockholders. This is important to identify features, functionalities and scope of the project. Several requirement gathering methods were used to select suitable methods. An overview of the factors that used to select each applied requirement elicitation method is given in Table 2.

<b>Method 1</b>	<b>Literature Review</b>
<p>Literature review is an important method in requirement elicitation task because it has rich resources such as IEEE and ScienceDirect. It can be useful to narrow down the scope the project, identify different approaches and its strength and weakness and requirement identification. Literature review is wide area, so it will be very valuable to gain knowledge and ideas. It has some drawbacks time consumption, delays in publications and vast amount of materials to go through. Even though it has some drawbacks compared to its benefits it still valuable, so this method selected.</p>	
<b>Method 2</b>	<b>Interviews</b>
<p>Interviews are great method for identifying valid requirements because it directly involves stockholders and through interviews can clarify doubts related to project, identifying business points of requirements and ensures achievements of the project feasible. Even though this method consumes a lot of time information gathered from this method valuable. Hence, this method is selected.</p>	
<b>Method 3</b>	<b>Online Questionnaire</b>
<p>Technical questionnaires focus on identifying technical requirements of project. When gathering information from many people through interview takes time and money. The survey insists the users to choose from the given options agree / disagree or rate something. So, it is suitable way to gather valuable information from many people with less work.</p>	
<b>Method 4</b>	<b>Requirement workshop</b>
<p>Requirement workshop is method of involving stockholders to communicate and identifying beneficial requirements. Even though this is a suitable method, it is not selected because unavailability of stakeholders and project's time limitation.</p>	
<b>Method 5</b>	<b>Brainstorming</b>
<p>Brainstorming with domain experts and group of people with knowledge in the area helps to gain knowledge.</p> <p>Even though it consumes long period, communication time and the corporation level are high, as each and every party can communicate and discuss openly since precious information is gathered.</p>	

<b>Method 6</b>	<b>Reverse Engineering</b>
Reverse engineering method used when system doesn't have enough documentation. This method isn't suitable because it takes more time and it required to study many technologies.	

*Table 4.2 - Requirement elicitation methods*

### 4.3.1 Literature survey

Literature review conducted on title generation and creative sentence generation domains to identify the research gap of the project and study about related studies in domains. To understand the related studies in depth, reviews of related studies classified into different groups, identified drawbacks and benefits and analyzed in relation project scope. Further, approaches to implement system studied and reviewed to identify suitable approach for system.

Literature survey can be found in [Chapter 3](#).

### 4.3.2 Interviews

Conducting an interview is the best manner to gather requirements from the stockholders. Limitation of interview in this project is, since the project outcome is a framework; People with knowledge in both domain level and technical level is suitable for interview but it's hard to find people in expert in both areas, interviews conducted on expertise in marketing and NLP/ software engineering.

#### 4.3.2.1 Execution of Interviews

Details about interviewees can be found in [Appendix B](#).

#### 4.3.2.2 Analysis of Interviews

<b>Interview - 1</b>	
<b>Interviewee</b>	Interviewee 2
<b>Interview type</b>	Face to face interview
<b>Purpose</b>	Identify requirements from content creator's perspective
<b>Findings</b>	<ul style="list-style-type: none"> <li>Enhance creative title generation with candidate titles and user will be given option to choose.</li> </ul>

	<ul style="list-style-type: none"> <li>• Generating just title from document isn't useful for content creator, they expect more features such as SEO optimization and grammar mistake findings.</li> <li>• If the framework supports like plugin to a system, it will be beneficial to automate the process.</li> </ul>
<b>Interview - 2</b>	
<b>Interviewee</b>	Interviewee 1
<b>Interview type</b>	Phone conversation
<b>Purpose</b>	Identify technical requirements for the framework (designs and features)
<b>Findings</b>	<ul style="list-style-type: none"> <li>• Suitable architecture designs for framework and pros and cons of designs.</li> <li>• Identified developer's perspective and expectations required from the framework.</li> <li>• Planning and development stages of framework.</li> </ul>

*Table 4.3 - Analysis of Interview*

### 4.3.3 Online Questionnaire

A technical questionnaire provided among software engineers who has experience at Aeturnum Lanka. Purpose of this questionnaire to identify,

- Suitable programming language for NLP.
- Functional & Non-Functional requirements from experts' perspective.

#### 4.3.4 Analysis of technical questionnaire

##### i. Which programming language is better to be used when working with NLP?

Intention of this question to identify suitable programming language for the project. More than 58% of participants who take part in the survey voted that Python programming language is better than other languages for working with NLP.

Which programming language is better to be used when working with NLP?

12 responses

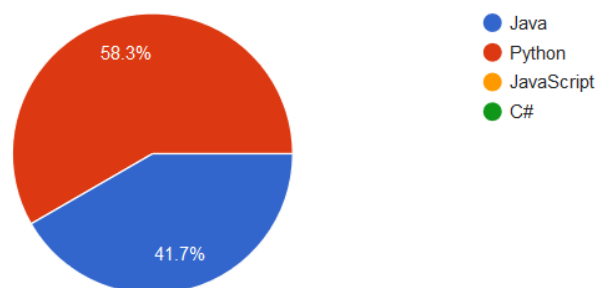


Figure 4.2 - Better programming language for NLP interview question & answers

##### ii. Why do you find above mentioned language better than others?

Intention of the question to identify features in selected programming language. Participants who voted for both Python and Java mentioned good documentation, community help and large numbers of frameworks and library support. Python voters mentioned that it is easy to learn and provides functional features compared to other languages and Java voters mentioned it provides type safety and rich OOP features. As per previous question results and suggestions from this question, Python programming language is identified as a suitable language for NLP by voters.

##### iii. Should framework provide methods for developers to implement their own logic?

This question asked to identify how important that is to provide methods for developers to implement their own logic. Almost all participants voted yes for this feature.

## Should framework provide methods for developers to implement their own logic?

12 responses

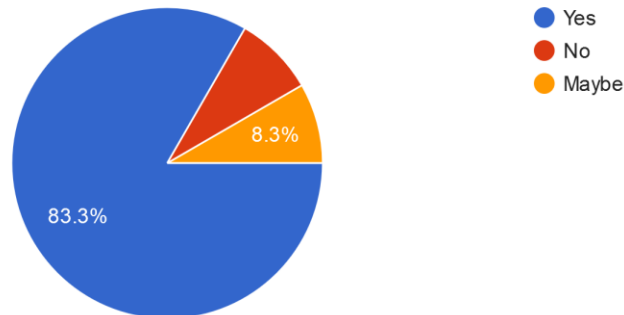


Figure 4.3 - Provide method for developers to implement interview question and answer

### iv. Should framework provide summary report for tasks?

Intention of this question to identify importance providing summary for tasks such as training creative text. Most participants voted yes, and few participants voted no and maybe. Based on the votes it is identified that providing summary reports (feedbacks) for task is important and required by developers.

## Should framework provide summary report for tasks?

12 responses

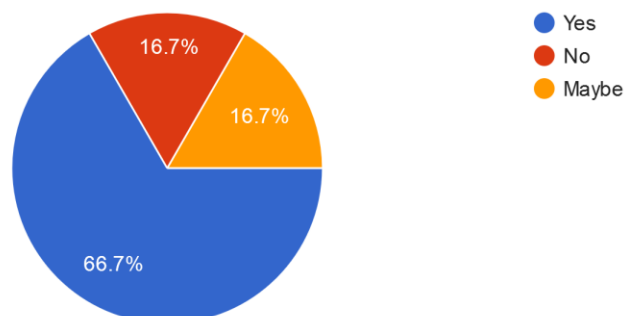


Figure 4.4 - Provide summary report interview question and answer

### v. What are the important requirements expected from this framework, marked down in scale of 1 to 5?

Intention of this question is to identify importance level of expected non-functional requirements from developer's perspective. Most participants gave 4 and 5 points to performance and accuracy. Next extendibility and usability have 2 and 3 votes in most. Based



on vote's summary participants expect performance and accuracy as main requirements and extendibility and usability as luxury requirements.

What are the important requirements expected from this framework, marked down in scale of 1 to 5?

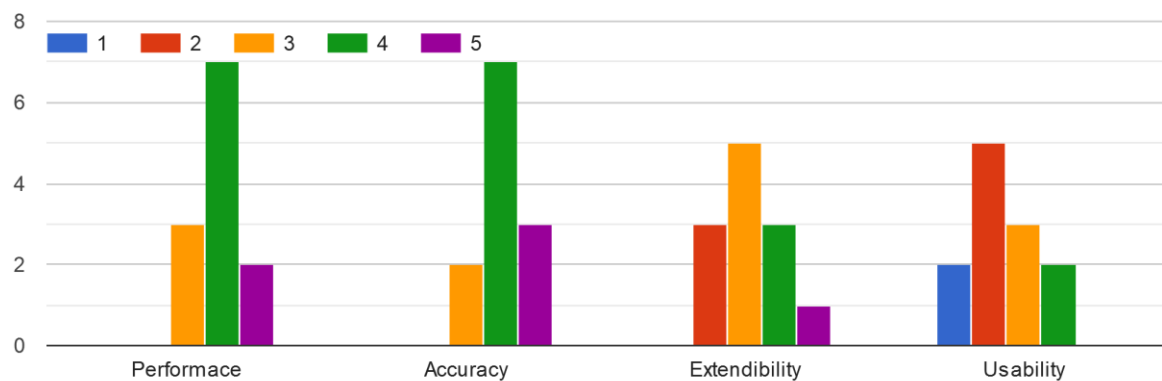


Figure 4.5 - Important requirement for framework interview question and answer

### 4.3.5 Findings derived from the Requirement Elicitation

The finding derived from the requirement elicitation process categorized in the table below.

#	Findings	LR	Interview	Questionnaire
1	The framework should have basic implementation of identification and generation		✓	✓
2	Should use selectional preference to effectively identify creative text	✓		
3	Use interpretation task to improve accuracy of matching creative text with keywords	✓		
4	Use of dependency parsing in selectional preference task	✓		
5	The framework should allow developers to make their own modifications in features		✓	✓

6	The framework should produce summary report for creative text identification task		✓	✓
7	The framework should include features such as SEO analysis and grammar validation.		✓	

*Table 4.4 - Findings derived from requirement elicitation*

## 4.4 Analysis & Design Methodologies

### 4.4.1 Design methodology

Object-oriented design methodology chosen for as suitable design methodology for the project. The comparison and reason discussed in [Section 2.7](#).

### 4.4.2 Modelling Language

SysML and UML are the identified major modelling languages. SysML is extension of a subset of UML and its suitable for project system engineering activities. On other hand UML language significantly designed for support software-based application hence, UML is identified as suitable modeling language.

## 4.5 Use Case Diagram

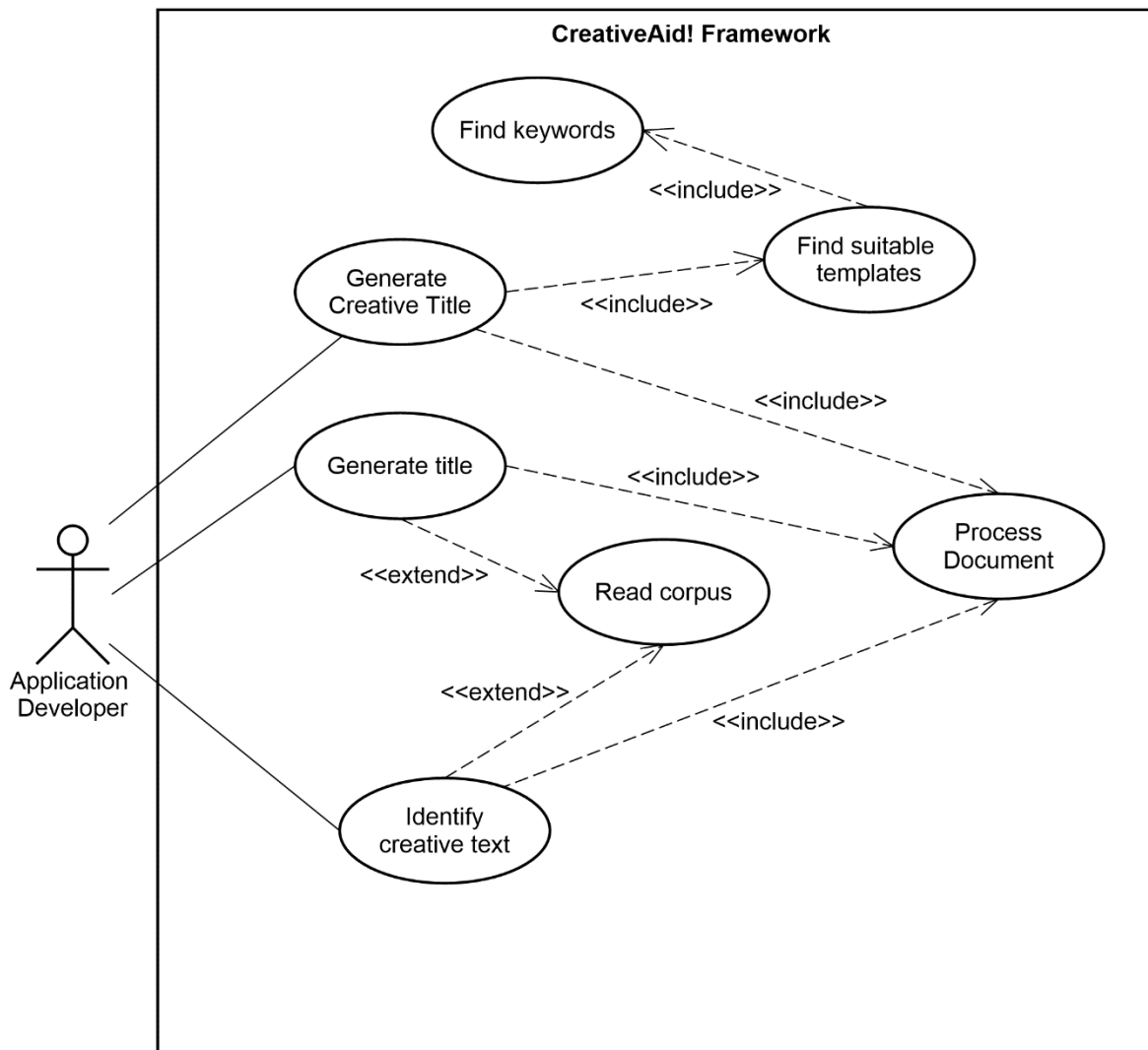


Figure 4.6 - Use case diagram

Identify creative text, generate title and generate creative title are the main three use cases directly interact with application developer. These use cases work independent. Generate Title and Identify creative text has extended use case read corpus. Read corpus use case is optional because the input for identify creative text and generate title can be directly given of can be read from corpus. Process document use case does the text processing and NLP related tasks.

### 4.5.1 Use Case Descriptions

An important use case description provided below. Rest of the use case descriptions documented in [Appendix C](#).

<b>Use Case ID</b>	UC1
<b>Use Case Name</b>	Identify Creative Text
<b>Priority</b>	High
<b>Participating Actors</b>	Application Developer
<b>Precondition</b>	<ul style="list-style-type: none"> <li>• Corpus or sentences list should be provided by developer</li> <li>• SP model should be loaded</li> <li>• Clustering model should be loaded</li> </ul>
<b>Postcondition</b>	Generate report of processed information
<b>Included Use Case</b>	Process Document, Interpret creative text
<b>Triggering event</b>	Application Developer execute identification task
<b>Description</b>	Application Developer is able to setup corpus or give sentences to framework then framework will return process identification text and process information report.
<b>Main Flow</b>	
<ol style="list-style-type: none"> <li>1. Get title and creative text templates and validate</li> <li>2. &lt;&lt;include&gt;&gt; Process document</li> <li>3. Extract verb, noun pairs</li> <li>4. Find clusters of pairs</li> <li>5. Calculate SPS score for word pairs</li> <li>6. Calculate SA score for word pairs</li> <li>7. Identify creative text score for word pair</li> </ol>	

<b>Alternative Flow</b>
<ul style="list-style-type: none"><li>• Alternative Flow 1 At step 1. If developer provide a corpus     &lt;&lt;Extend&gt;&gt; Read Corpus and go to step 1</li><li>• Alternative Flow 2 At step 2. If corpus preprocess error encountered     Display error message</li></ul>
<b>Exceptional Flow</b>
<ul style="list-style-type: none"><li>• Exceptional Flow 1 In step 4,     Clusters for word pairs not found     Display error message continue to next sentence</li><li>• Exceptional Flow 2 In step 5,     SPS calculation returns empty     Display error message and continue to next sentence</li><li>• Exceptional Flow 3 In step 6,     SA calculation returns empty     Display error message and continue to next sentence</li></ul>

*Table 4.5 - Use case description*

## 4.6 Activity Diagram

Framework has two activity flows so, two activity diagrams provided for better understanding. Activity diagram for creative title generation and creative text identification.

### 4.6.1 Activity Diagram for identification part of framework

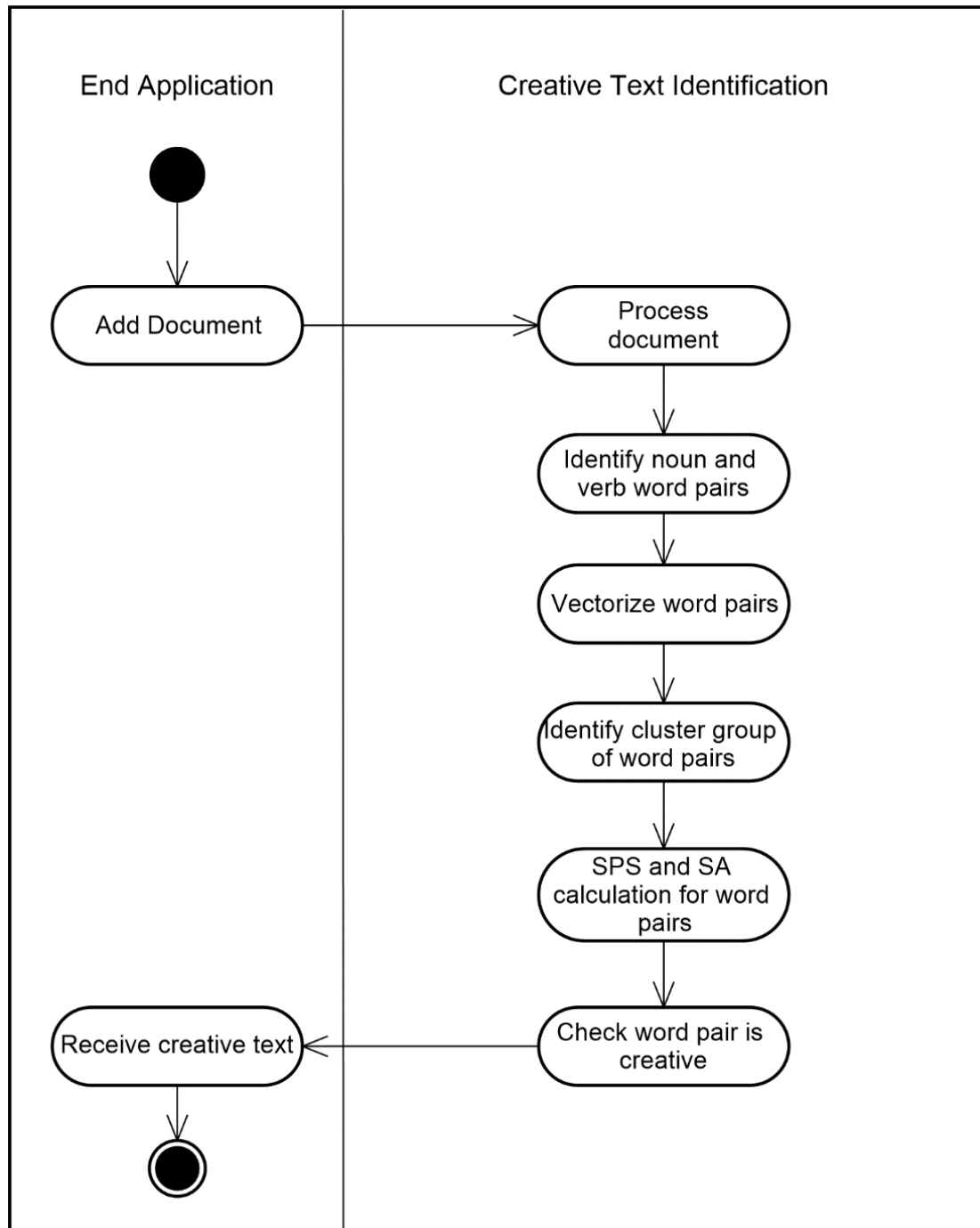


Figure 4.7 - Activity diagram for identification

### 4.6.2 Activity diagram of creative title generation

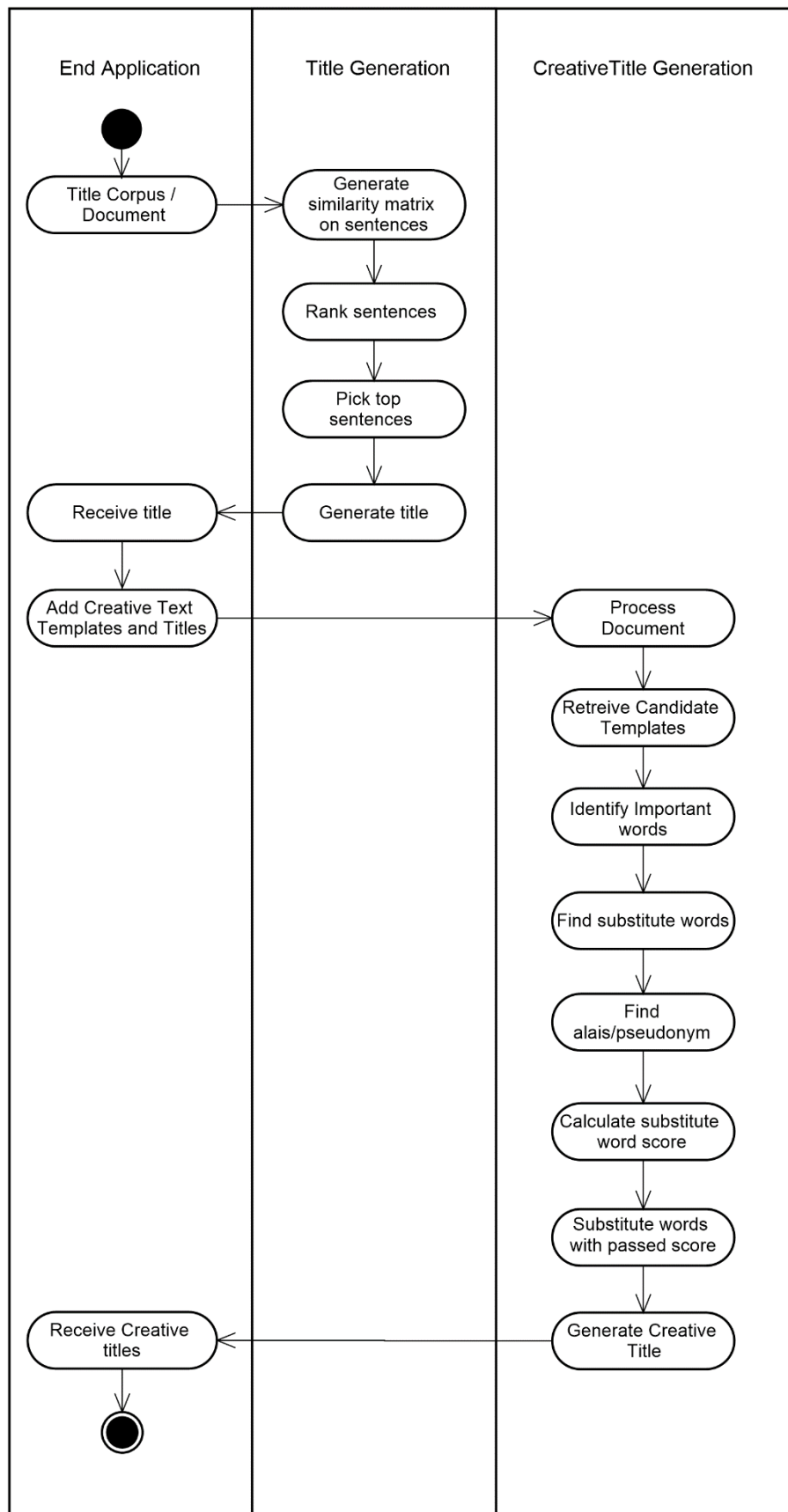


Figure 4.8 - Activity diagram for generation

## 4.7 Functional Requirements

Considering resource and time constraints, it is difficult to implement all requirement hence, identified requirements in elicitation process prioritized as follow,

- Critical (C) – These requirements are represented as core functionalities of the system.
- Important (I) – Requirements that are considered to be not critical, but they are important to system.
- Luxury (L) – Requirements which are proposed to implement in further development.

Identified functional requirements of framework represented in Table 3.

#	Requirement	Priority	Use case(s)
FR1	<b>Read corpus and extract contents</b> If input for framework given in corpus format framework should be able to process it with corpus reader object.	C	Read Corpus
FR2	<b>Process documents</b> The framework should provide NLP features and text preprocessing features to convert document to required manner.	C	Process Document
FR3	<b>Identify creative sentences</b> The framework should identify creative sentences in corpus and sentences with selectional preference method.	C	Identify Creative text
FR4	<b>Generate title</b> The framework should provide basic features to generate titles from corpus and sentences for developers and it should be extended to make own changes.	C	Generate title
FR5	<b>Find suitable templates for creative title</b> Before task of creative title generation framework should able to identify suitable templates for creative title.	C	Find suitable templates



FR6	<b>Find important words in title and templates</b> The framework should be able to identify important words in title and template to for substitution task.	C	Find important words
FR7	<b>Identify alias and pseudonym names</b> The framework should be able to identify alias and pseudonym names of things, objects and people then map identified details with creative text information.	I	Identify Creative text
FR8	<b>Generate creative title</b> Framework should be able to generate creative title from title and creative templates given by developers. Also, framework should provide methods for developers to make own modifications.	C	Generate creative title
FR9	<b>Map alias/pseudonym with creative title</b> In the process of generating creative title, framework should be able to identify alias/pseudonym for keywords	I	Generate creative title
FR10	<b>Validate generated creative title</b> Once the creative title generated framework should be able to validate title is grammatically valid and in correct format.	L	Generate creative title
FR11	<b>Check SEO analysis</b> Once the final candidate creative title generated framework should be able to analyze SEO of titles and produce a report.	L	Generate creative title

Table 4.6 - Functional requirements

## 4.8 Non-Functional Requirements

Identified non-functional requirement of the framework represented in Table 4.

#	Requirement	Priority	Description
NFR1	Accuracy	C	The framework should accurately identify and generate creative text.

NFR2	Performance	I	The identification (training) part of framework should work without delay because when developer might setup large corpus.
NFR3	Extendibility	L	The framework must to be extendable for developers to make additional adjustments and enhancements with their own implementation.

*Table 4.7 - Non-functional requirements*

## 4.9 Scope Refinement

- During the requirement elicitation process it was identified that developers expect minimal abstraction level of implementation hence, the framework should provide basic implementation for identifying creative text in document and generating creative title in a way a developer can extend it and make his own implementation.
- The identification part of the framework should be implemented as optional for developers hence framework should contain a pre-attached creative text data set for generate creative title.
- Since the project objective is to train and generate title, no security requirements will be considered.
- Critical level of functionalities will be implemented. Important (I) functionalities will be implemented only if time permits. Luxury (L) functionalities will be developed in the future versions of the framework.

## 4.10 Chapter Summary

The chapter focused on identifying stakeholders and requirements from the stakeholders using appropriate requirement elicitation techniques. Identified stakeholders and their roles are described with the help of an onion diagram and a description table. Standard requirement elicitation techniques are compared. Questionnaires, interviews and literature reviews the techniques used to identify the requirements of framework. Object Oriented Analysis is chosen as the suitable modeling technique for this system. Therefore, based on the requirements, a use case diagram is produced of the framework to the stakeholders. Then the functional and non-functional requirements of the framework are identified and prioritized. Finally, the system scope was refined.

# **Chapter 5: Design Specification**

## 5. Design Specification

### 5.1 Chapter Overview

The chapter describes the architecture and design of the project. Chapter begins with discussing design goals of the framework and continues to discuss about high and low-level design topics of the framework. Finally, the chapter critically analyze and evaluate design goals how they achieved through these designs.

### 5.2 Design Goals

Design goals of the system is based on non-functional requirements gathered from [Section 4.8](#) in system requirement specification chapter. These goals followed during the design phase of the proposed system.

#### Easy Integration

Since the project outcome is a framework developer should be able to integrate the framework in client's application without any complex configuration with few steps. The framework has main two separate parts identification and generation both parts should be understandable for developer.

#### Modularize components

Components of the frameworks should be modularized to help the developer to easily import the module and extend.

#### Loose Coupling

Framework's component or module should focus on one particular task only and should be developed as not heavily depending on other components. In this way components can be changed without affecting the whole system.

### 5.3 Architecture Styles

Adapting suitable architecture styles enhances to promote and partitioning the design reuse to overcome common programming issues. To identify the suitable architecture styles for system multiple architecture styles advantages and disadvantages of discussed based on project's purpose.

#### Client-server architecture

The framework is not a type of client server pattern because It doesn't have two parties to communicate with each other.

### **Domain Driven Development**

To develop domain driven development architecture, the domain has to be studied in depth. Also, this architecture style requires lots of resource. The framework is not too complicated to design a domain specific language also framework developed by a single developer hence, this architecture style is identified as not suitable.

### **Object-Oriented Architecture**

Framework's components will be implemented using object-oriented architecture. The main reasons for selecting this are easy implementation and past experience in OOA. The framework requires the object-oriented principles which make ease of the implement of each and every component.

### **Modular Architecture**

Modular Architecture, as a style, helps us view the system, not just in layers or services, but goes one level below as composition of smaller, physical modules. This architecture style suitable because this framework can be decomposed to different groups of modules. The framework can be broken into three modules identifier, generator and nlp modules. This helps framework to decouple subtasks and changes can be made without affecting other modules.

### **Pipeline Architecture**

This architecture decomposes a task that performs complex processing into a series of separate elements that can be reused. It contains filters, that transform or filter data before passing to next component. This style suitable for identification task of framework because it goes series of tasks tagging, tokenizing, parsing, text identification.

## 5.4 High Level Architecture of the Framework

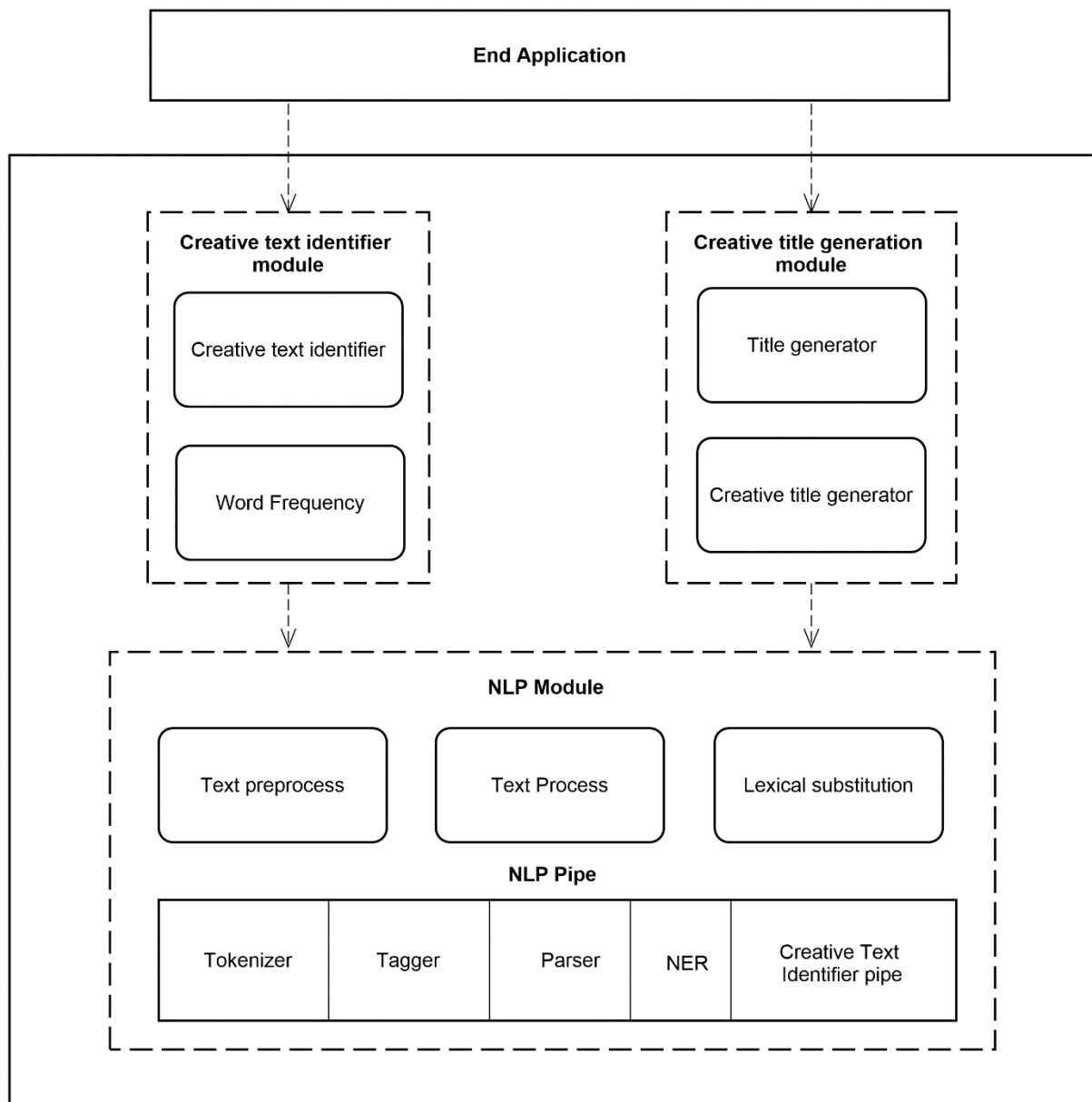


Figure 5.1 - High Level Architecture Diagram

The framework's components separated to different modules and layers to improve the understating and maintainability of the project.

### 5.4.1 Modules of framework

This framework contains three modules they are identifier, generator and NLP modules. Modules and their work discussed below,

#### Identifier module

This module is responsible for retrieve sentences directly or via a corpus and identify creative sentence in corpus.

**Generator module**

This module generates titles from document and pass to generate creative titles from templates and titles.

**NLP module**

This module retrieve document and perform NLP (tokenizing, tagging, parsing and more) and basic text preprocessing. NLP module used by both other modules.

**5.5 Low level design models**

Low level design stage is the stage where the internal logical design of the actual program code is reflected. The low-level designs based on OOP methodology as mentioned in SRS chapter. Low level designs are followed by the high-level designs. Low level diagrams of the project are class diagram, sequence diagram and context diagram.

### 5.5.1 Class Diagram

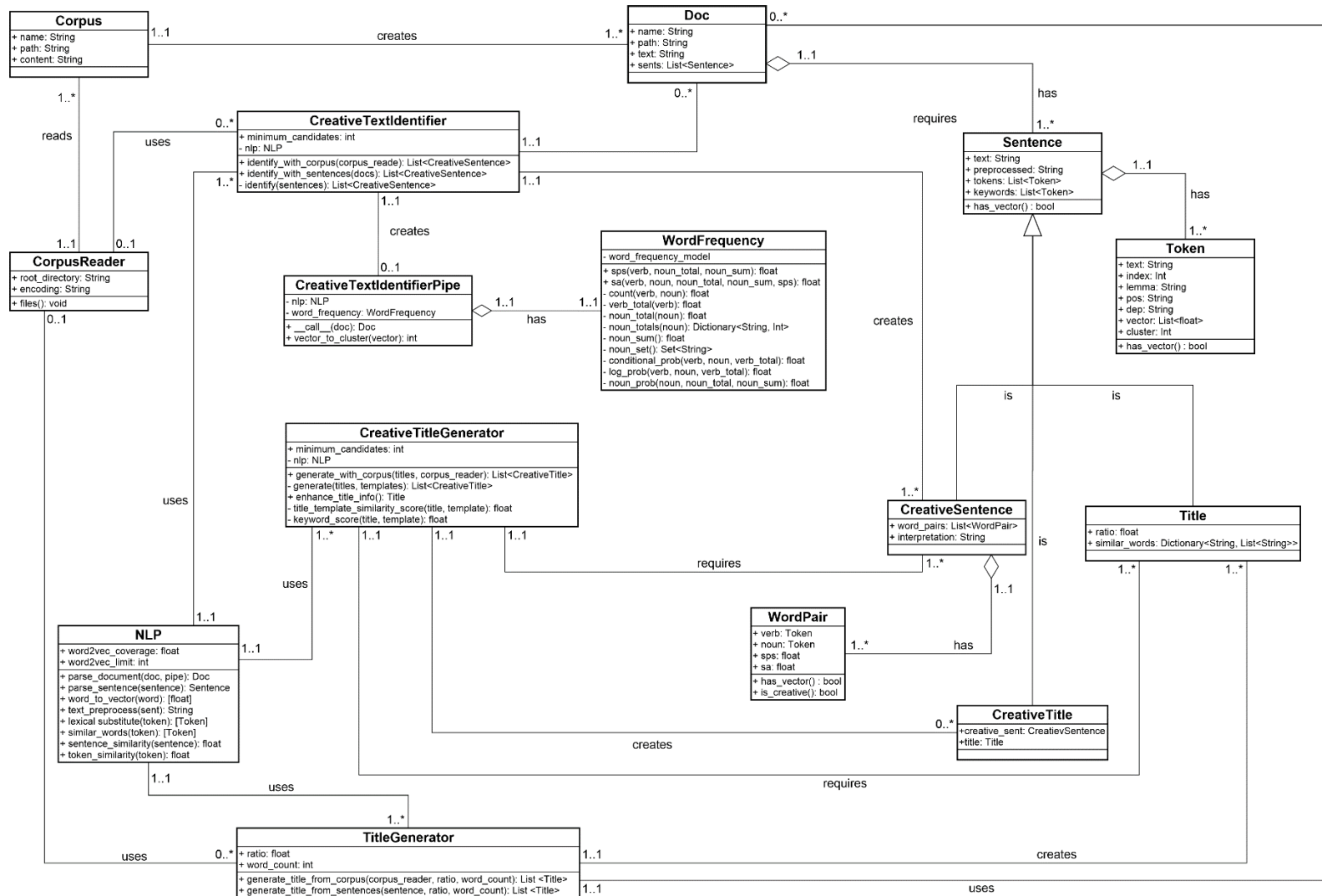


Figure 5.2 - Class diagram



### 5.5.1.1 Class diagram description

Class Name	Description
Corpus	Represents the corpus.
Doc	Represents the document in the corpus.
Sentence	Represents a sentence in document. Attribute preprocessed is sentence converted for nlp tasks and keywords are tokens which are identified as important.
Token	Represent word and token in sentence.
WordPair	Represents creative words in a sentence. Contains noun and verb token and information related to these to relation in selectional preference.
CreativeSentence	Represents creative sentence identified by CreativeTextIdentifier.
Title	Represents title generated from document by TitleGenerator.
CreativeTitle	Represents creative title generated by CreativeTitleGenerator.
NLP	Process NLP and text related tasks.
Corpus Reader	This is used to read corpus and return Corpus object.
WordFrequency	This class responsible for SPS and SA calculation for creative text identification task. It requires word frequency model.
CreativeTextIdentifier	This class reads sentences from given doc or corpus reader and identifies creative text. This class is responsible for generating CreativeSentence class.
TitleGenerator	It takes sentences or corpus class and generates titles for it.
CreativeTitleGenerator	This class takes titles and creative sentences objects to generate CreativeTitle. This is class contains number of subtasks such as lexical substitution, title enchantment, similarity calculation and more.

Table 5.1 - Class diagram description

### 5.5.2 Sequence Diagram

A sequence diagram explains the interaction between the object in sequential manner. Sequence diagrams were designed only for core components of the framework. Rest of the sequence diagrams can be found in [Appendix D](#).

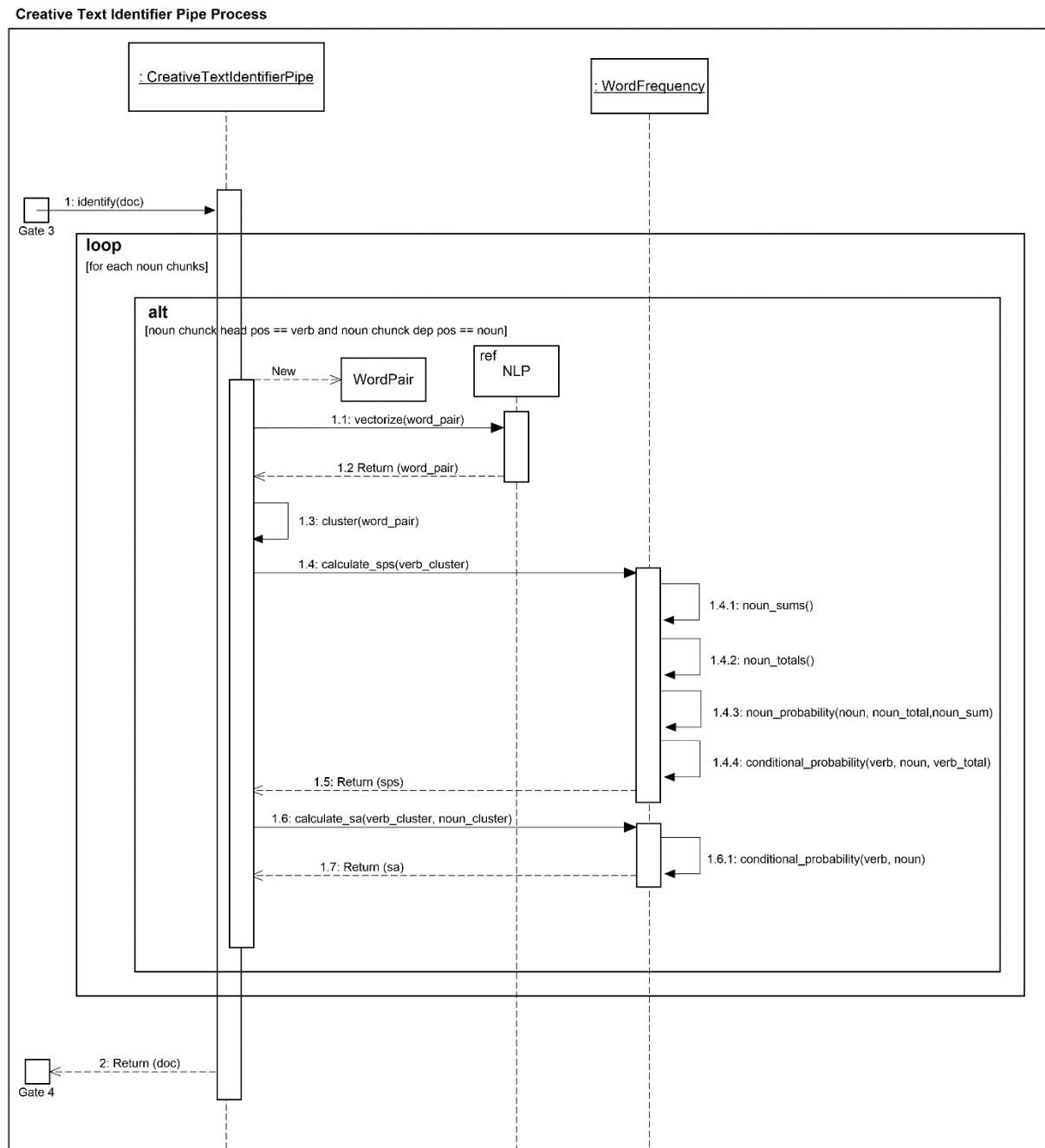


Figure 5.3 - Sequence diagram for creative text identification pipe

### 5.5.3 Context Diagram

Context diagram shows the relationship between system and other or external entities. It defines the scope and boundaries of a system at a glance including the other systems that interface with it.

The important interactor with the framework is developer when the framework extended to an application this role could be broken in different roles. As recommended order, first developer should provide corpus or raw sentences for creative identification task and framework returns creative sentences in the corpus. Second, developer should provide corpus of documents (body part of articles, book, news or more) or raw sentences to framework and framework generate generates titles for document. Third, developer should provide generated titles and creative text to framework and it returns creative titles. During generating creative titles framework retrieves alias and pseudonym details for keyword from WikiData (Kinzler and Pintscher, 2014).

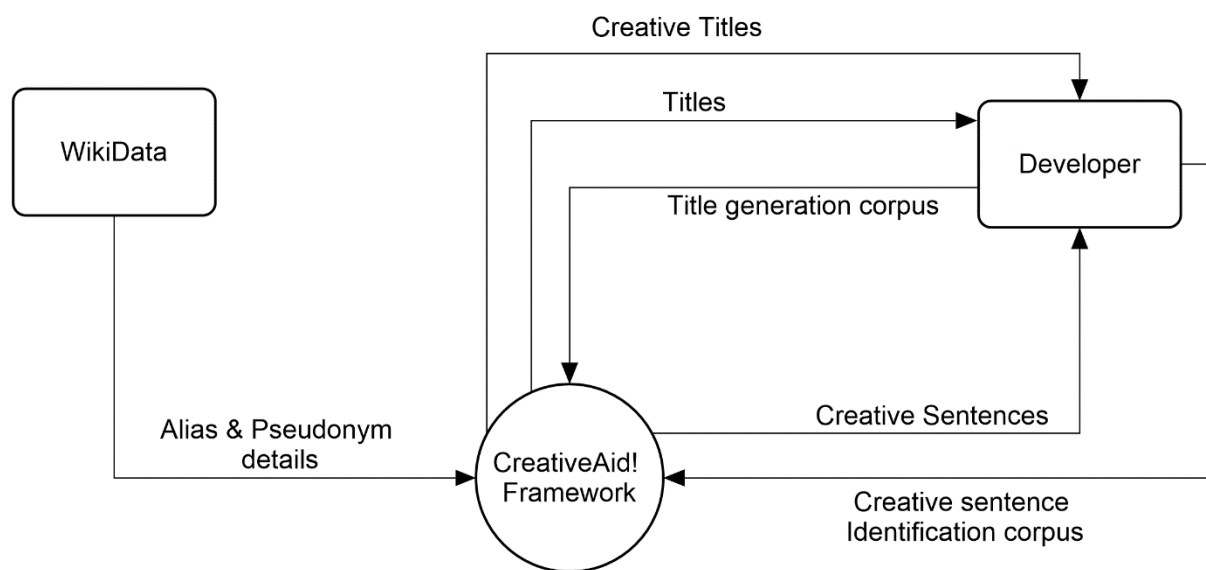


Figure 5.4 - Context diagram

## 5.6 Chapter Summary

This chapter discussed the architecture and the design of the framework. The chapter started with identifying the architectural goals. The different architectural styles were analyzed, and suitable architectural styles were selected to satisfy the architectural goals. The high-level design and architecture of the proposed solution was then described. Finally, the chapter presented low-level design diagrams, class diagram and sequence diagram. Also, context diagram provided to shows the relationship between framework and other entities.

# **Chapter 6: Implementation**

## 6. Implementation

### 6.1 Chapter Overview

The chapter discusses the selection of tools and technologies. It presents a critical evaluation of tools and technologies and then provides justifications for the selections. The chapter also discusses the features of the framework and elaborates how each feature of the framework is implemented using the libraries. It also presents code snippets to give a clear picture of the implementation.

### 6.2 Technology selection

#### 6.2.1 Programming language selection

Selection of suitable programming language is critical task. Based on developer's past experiences and programming language popularity in NLP, Java and Python identified as suitable for the project. Java and Python both general purpose languages and has vast community support and both support OOP design methodology. But comparing Java with Python, Python widely used in many research projects and supports lot of NLP libraries. Library available for Java are mostly developed for research purposes (CoreNLP) and they are slow. On other hand Python supports rich industrial standard libraries such as spaCy and Text Blob, which are faster and have produce more accurate results. Based on the review and questionnaire done [Section 4.3.3](#) Python choses as suitable programming language for project.

#### 6.2.2 Libraries Selection

One of the important tasks of the framework is to parse given document. Therefore, it is important that considering this an important factor in library selection. Based on popularity and recommendation spaCy, NLTK and Sandford CoreNLP identified as important libraries for NLP.

Comparison	spaCy	NLTK	Sandford CoreNLP
Python support	✓	✓	✓
Tokenization	✓	✓	✓
Part of the speech tagging	✓	✓	✓
Sentence segmentation	✓	✓	✓

Dependency parsing	✓	✗	✓
Time Consumption	Medium	High	High
Resource Consumption	High	High	High
Accuracy	High	Medium	Medium
Documentation and online community	Medium	High	Medium

Table 6.1 - Library selection comparison

Based on this finding, it was decided to short down the libraries to Sandford CoreNLP and spaCy because NLTK doesn't support dependency parsing which is an important task of the framework. When comparing spaCy and CoreNLP, spaCy benchmarked as fastest syntactic parser in the world with highest accuracy of 91.8 % (Choi, Tetreault and Stent, 2015). Since the framework has creative text identification, developers might train huge size of corpus so, framework should able process it faster with accurate results. CoreNLP library implemented for academic purposes to try out different things but spaCy is an industry level library, every aspect of spaCy chosen considering a production level library. Based on this information spaCy chosen as suitable NLP library for the project.

### 6.2.3 Word Embeddings Selection

Word embeddings used for different purposes in the framework. Selection of word embedding discussed in LR [Section 3.6.1](#). Glove embeddings for clustering model because SP and Glove both are count based features and Word2vec used for getting similar words in creative title generation task.

### 6.2.4 Clustering library selection

In the task of SPS clustering is required to generalize words into groups. TensorFlow and scikit-learn are the popular libraries for text clustering. In recent years scikit-learn library widely used by researchers because of it fast parallel process optimization in CPU. TensorFlow is a GPU optimized library for deep learning work. Both libraries have wide community support but in TensorFlow provides low level support for implementing on other hand scikit-learn is a high-level library that contains implementations of several machine learning algorithms, so users can define a model object in a single line or a few lines of code. Since clustering implementation doesn't require custom implementation sci-kit learn is suitable. Based on these reasons scikit-learn chosen for k-mean clustering library.

## 6.3 Tools selection

### 6.3.1 IDE selection

	BEST PYTHON IDES OR EDITORS	PRICE	MULTI LANGUAGE SUPPORT	CROSS PLATFORM
91	 PyCharm	£6.90 - £49.90 /MONTH	Yes	Windows, macOS, Linux, FreeBSD
88	 Visual Studio Code	-	Yes	Yes
88	 Vim	FREE	Yes	Yes
83	 Sublime Text	\$80	Yes	Yes
82	 Spacemacs with Python layer	-	Yes	Yes
--	 Jupyter	-	-	-
74	 Spyder	FREE	No	Yes

Figure 6.1 - IDE comparison (Slant, 2019)

There variety of IDE available for python development Spyder, Pycharm, Atom, Jupyter Notebook and more. Amon these IDEs Pycharm was chosen for development because past experience with Jetbrain IDEs, and the experience is same with Pycharm also it has many features such as powerful debugger, inbuilt VCS integration, plugins and auto completion.

### 6.3.2 Version Control System selection

Version Control System (VCS) is a most important tool for software development. A VCS manages and tracks changes to documents, files, codes and other collections of information. Github and Bitbucket are the popular version control systems available today. Both have many features unlimited storage, ease of usage and private repositories but Github is stable it has been in business in early stages hence, Github chosen as suitable VCS for development.

## 6.4 Data and Models preparation

The framework requires pretrained models and data to achieve identification and generation tasks. Model trained and data gathered from different source and it stored in pickle format and at runtime pickle loaded to framework.

### 6.4.1 Clustering model

In order to identify a cluster of a word set of pretrained (clustered) model is prepared. For the clustering model [Section 3.5](#) decided to use k-means clustering but k-means clustering requires lengthy time. Due to limited time a slightly different version of k-means, minibatch k-

means clustering used for clustering. Minibatch k-means is a variant of k-means algorithm which uses mini-batches to reduce the computation time (Scikit-Learn Mini-Batch K-Means, 2019).

Word embedding for clustering is decided to use Glove word embeddings (Pennington, Socher and Manning, 2014) with 300 dimensions, trained on 840B word corpus. Embeddings grouped into 200000 clusters with k-mean++ initialization, a maximum of 300 iterations. The model saved in pickle format and loads into framework at runtime.

Example of clustered word groups of verbs.

<p>give, put, do, want, call, bring, take, ask, go, get, shoot, save, jump, watch, forget, stop, kill, miss, move, decide, hang, please, wait, hurry, shot</p> <hr/> <p>realize, discover, adapt, anticipate, promise, communicate, perceive, abandon, intend, discern, solve, fear, endure, trust, comprehend, overcome, concentrate, embrace, recognize, realize, reap, fathom, accomplish, insist, dedicate, pretend</p> <hr/> <p>complete, offer, include, require, enter, use, express, obtain, order, base, counter, double, avoid, vary, advance, suit, value, reverse, scale, account, direct, free, block, further, single, level, condition, frame, board, label, side, chain, appropriate, number, except</p>
--

*Figure 6.2 - Verb cluster groups*

Example of clustered word groups of nouns.

<p>you, them, me, wait, call, miss, go, taking, move, let, do, watch, stop, shot, shoot, ask, get, take, hurry, jump, save, hang, want, bring, shooting</p> <hr/> <p>it, people, place, deal, one, they, because, lot, chance, fine, trouble, encounter, past, scene, impression, worth, rest, fortune, appearance, plenty, couple, surprise, prospect, mine, find, spot, while, might, buzz, happening, finding, still</p> <hr/> <p>interest, recognition, knowledge, subject, purpose, foundation, relation, appeal, term, validity, prestige, obligation, context, nature, understanding, aspect, insight, merit, importance, basis, consideration, necessity, emphasis, authority, significance, representation, nature</p>
---

*Figure 6.3 - Noun cluster groups*



## 6.4.2 Selectional Preference data

Selection preference information can be gathered from corpus. BNC corpus (Hawtin, 2018) and Wikipedia are identified as popular corpus for selectional preference. When comparing size of corpus Wikipedia corpus contains vast amount of text about 1.6 billion words and BNC has limited amount of words. The drawback with Wikipedia corpus is its only contain only one genre (Haagsma and Bjerva, 2016). Since the framework is built based on domain independent manner Wikipedia corpus isn't suitable. Hence, BNC corpus chosen for gather selectional preference information.

Using dependency parser of spaCy library, sentences in BNC corpus were parsed and all noun and verb pairs with the labels *nsubj* and *dobj* were extracted. Extracted words were lemmatized to increase the identification coverage.

```
for doc in nlp.pipe(clean_sentences):
    for chunk in doc.noun_chunks:
        if chunk.root.head.pos == VERB:
            if chunk.root.dep == nsubj:
                word = chunk.root.text if chunk.root.pos == pron else chunk.root.lemma_
                verb_nsubj.append(word + "," + chunk.root.head.lemma_)
            elif chunk.root.dep == dobj:
                word = chunk.root.text if chunk.root.pos == pron else chunk.root.lemma_
                verb_dobj.append(word + "," + chunk.root.head.lemma_)
```

Figure 6.4 - Selection preference data gathering

Once the pairs collected it saved into a text file, then they get vectorized using Glove and cluster group identified using clustering model. Then word pairs grouped into count frequency (that how many times a verb and noun occurred in corpus).

```
def get_verb_noun_frequency(word_pairs):
    verb_freq = {}
    for word_pair in word_pairs:
        verb = int(word_pair[1])
        noun = int(word_pair[0])

        if verb not in verb_freq.keys():
            verb_freq[verb] = {}

        if noun in verb_freq[verb].keys():
            verb_freq[verb][noun] += 1
        else:
            verb_freq[verb][noun] = 1

    return verb_freq
```

Figure 6.5 - Selection preference word pair normalizing

This frequency saved as pickle format and loaded into program in runtime. SPS and SA score calculated using the frequency.

### 6.4.3 Keywords selection model

In creative title generation part keywords in words for title and creative text should be extracted before substitute task. Three types of concepts derived during this task i) named entities, ii) important keywords, iii) similar and relevant words of keywords.

The important value of each word calculated based on the number of times that a word's lemma appears in the BNC corpus (Hawtin, 2018), it divided by the total number of title occurring in the corpus (i.e. the probability of the lemma). Important value above a certain threshold value are considered as key words. Named entities also included in keywords.

## 6.5 Technology stack

The technologies used to implement the modules components of the framework.

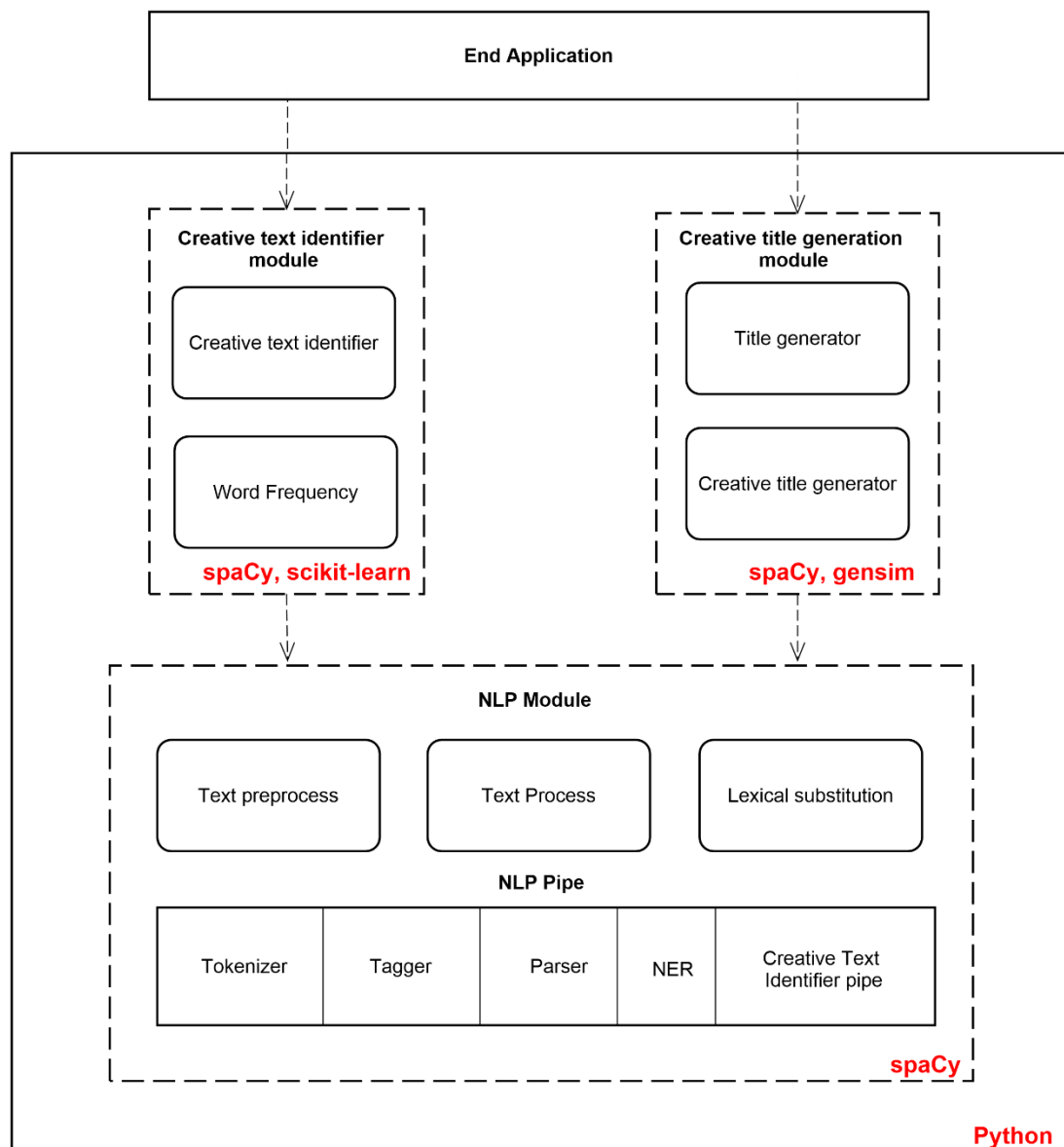


Figure 6.6 - Technology stack

## 6.6 Implementation of Functional requirements

Core functionalities of the framework described in this section with code snippets.

### 6.6.1 Receive input as raw sentences or corpus format

Sentences for title generation and identification tasks can be passed directly passed to system or can be passed via a corpus reader class. This takes root path of directory and read files in directory.

```
class CorpusReader:
}   def __init__(self, root, fields, encoding=None):
}       self.root = root
}       self.fields = fields
}       self.encoding = encoding
}
}   def corpus(self):
}       return self.CorpusIterator(self, self.corpus_files())
}
}   def corpus_files(self):
}       file_col = []
}       for root, dirs, files in os.walk(self.root):
}           for name in files:
}               file = Corpus(name, os.path.join(root, name))
}               file_col.append(file)
}       return file_col
}
}   class CorpusIterator:
}       def __init__(self, corpus_reader, files):
}           self.current = 0
}           self.length = len(files)
}           self.corpus_reader = corpus_reader
}           self.files = files
}
}       def __iter__(self):
}           return self
}
}       def __next__(self):
}           if self.current >= self.length:
}               raise StopIteration
}           else:
}               file = self.files[self.current]
}               file.contents = file.get_contents()
}               self.current += 1
}               return file
}
}
```

*Figure 6.7 - Corpus reader implementation*

Classes get the corpus reader object and get contents of at runtime. The CorpusIterator class used when corpus reader gets used by array function to read on by one.

### 6.6.2 Process documents

A set of subtasks for both generator and identifier modules are provided from here. Tasks such as clean sentences (remove words not related to process), parse sentences, vectorized, similar word identification and more can be related to process documents functionality.

```
def clean_sentence(self, sentence):
    sentence = sentence.lower()
    sentence = ''.join([i for i in sentence if not i.isdigit()])
    sentence = ' '.join([word for word in sentence.split(' ') if word not in STOP_WORDS])
    sentence = sentence.translate(str.maketrans('', '', punctuation))
    return sentence.strip()
```

Figure 6.8 - Clean sentences implementation

```
def unique(self, iter):
    """removes duplicates from iterable preserving order"""
    result = list()
    seen = set()
    for x in iter:
        if x not in seen:
            seen.add(x)
            result.append(x)
    return result
```

Figure 6.9 - Identify unique implementation

```
def process_candidates(self, candidates, target):
    """
    words to lower case, replace underscores, remove duplicated words,
    filter out target word and stop words
    """
    filterwords = STOP_WORDS + [target]
    return self.unique(filter(lambda x: x not in filterwords,
                             map(lambda s: s.lower().replace('_', ' '), candidates)))
```

Figure 6.10 - Process candidate implementation

### 6.6.3 Creative sentence identification

The creative sentence identification begins with passing a sentence list. This identification task works as pipeline. CreativeTextIdentificationPipe will be added spaCy's pipes array at last index. spaCy calls pipe one by one and finally call identification pipe.

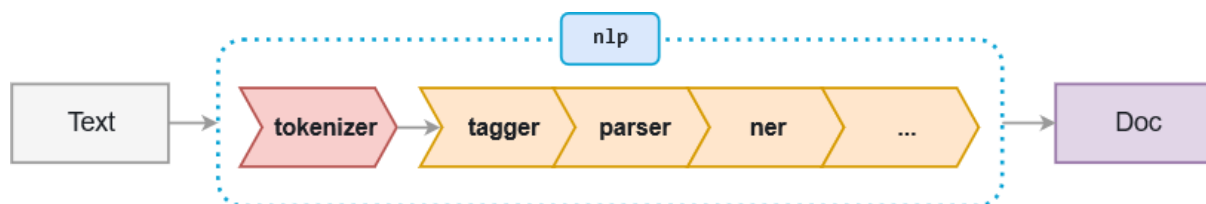


Figure 6.11 - spaCy pipeline example (Explosion AI, 2019)

Identification pipe gets noun chunks in parsed sentences then checks the word pairs of verb and noun (subject or object) if the suitable pairs found it'll go through series of tasks applied to lemmatization, vectorized, clustered into groups, SPS calculated and finally SA score calculated. Then it will be added to Span class. Finally, CreativeTextIdentification class gets the sentences by checking word pairs SA score checked against a threshold optimal value of creative text level If word pair SA value less than threshold,

```

class CreativeTextIdentifierPipe(object):
    name = "creative_text_identifier" # component name, will show up in the pipeline

    def __init__(self, nlp):
        self.nlp = nlp
        self.word_freq = WordFrequency(pickle.load(open(word_freq_path, 'rb')))
        # Register attribute on the Span. We'll be overwriting this on __call__
        Span.set_extension("word_pair", default=None)

    def __call__(self, doc):
        NSUBJ = 429
        DOBJ = 416

        for chunk in doc.noun_chunks:
            if not chunk.root.head.pos == VERB or not (chunk.root.dep == NSUBJ or chunk.root.dep == DOBJ):
                continue

            noun_norm = chunk.root.text if chunk.root.pos == PRON else chunk.root.lemma_
            noun = Token(noun_norm, chunk.root)
            verb = Token(chunk.root.head.lemma_, chunk.root.head)
            word_pair = WordPair(verb, noun)

            # word pair vectorized
            word_pair = self.nlp.w2v(word_pair)

            if not word_pair.has_vector():
                continue
            # word pair clustered
            word_pair = self.v2c(word_pair)

            # SPS identification
            word_pair.sps = self.word_freq.sps(word_pair.verb.cluster)

            # SA identification
            word_pair.sa = self.word_freq.sa(word_pair.verb.cluster, word_pair.noun.cluster)
            chunk._.set("word_pair", word_pair)
        return doc

```

Figure 6.12 - Creative text identifier pipe implementation

In order to increase the vectorizing coverage of words (there are chances a vector of a word cannot be found) the most similar word vector is returned when a specific word vector is not found.

```

def w2v(self, word):
    """
    Identify vector or most similar vector for given word
    :param word: word
    :return: vector
    """
    try:
        vector = self.word2vec.wv.get_vector(word.text_)
    except KeyError:
        try:
            similar = self.word2vec.most_similar(word.text_, topn=1)
            vector = self.word2vec.wv.get_vector(similar[0][0])
        except KeyError:
            vector = None
            logging.error(f"word_vec & similar not found for {word}")

    return vector

```

Figure 6.13 - Vectorizing implementation

## 6.6.4 Identify alias and pseudonym names

For this task a knowledge based Wikidata (Kinzler and Pintscher, 2014) is used. From Wikidata nick name, related name and other names are retrieved and used as alias and pseudonym names for named entities.

### 6.6.5 Generate Title

As task in creative title generation when user desired to generate creative title from contents of document this function generate title from contents by summarizing the contents into length or ratio to contents. The process begins with building similarity matrix on sentences with removing stop words. Then using PageRank scores for each sentence. Then based on the score and desired length top scored sentences will be generated as title.

```
def _generate_title(self, sentences, ratio, word_count, top_n=5):
    summarize_text = []

    # Generate Similarity Matrix across sentences
    sentence_similarity_matrix = self.build_similarity_matrix(sentences, STOP_WORDS)

    # Rank sentences in similarity matrix
    sentence_similarity_graph = nx.from_numpy_array(sentence_similarity_matrix)
    scores = nx.pagerank(sentence_similarity_graph)

    # Sort the rank and pick top sentences
    ranked_sentence = sorted(((scores[i], s) for i, s in enumerate(sentences)), reverse=True)

    for i in range(top_n):
        summarize_text.append(" ".join(ranked_sentence[i][1]))

    return " ".join(summarize_text)
```

Figure 6.14 - Generate title implementation

```
def sentence_similarity(self, sent1, sent2, stopwords=None):
    if stopwords is None:
        stopwords = []

    sent1 = [w.lower() for w in sent1]
    sent2 = [w.lower() for w in sent2]

    all_words = list(set(sent1 + sent2))

    vector1 = [0] * len(all_words)
    vector2 = [0] * len(all_words)

    # build the vector for the first sentence
    for w in sent1:
        if w in stopwords:
            continue
        vector1[all_words.index(w)] += 1

    # build the vector for the second sentence
    for w in sent2:
        if w in stopwords:
            continue
        vector2[all_words.index(w)] += 1

    return 1 - cosine_distance(vector1, vector2)
```

Figure 6.15 - Sentence similarity implementation

```

def build_similarity_matrix(self, sentences, stop_words):
    # Create an empty similarity matrix
    similarity_matrix = np.zeros((len(sentences), len(sentences)))

    for idx1 in range(len(sentences)):
        for idx2 in range(len(sentences)):
            if idx1 == idx2: # ignore if both are same sentences
                continue
            similarity_matrix[idx1][idx2] = self.sentence_similarity(sentences[idx1], sentences[idx2], stop_words)

    return similarity_matrix

```

Figure 6.16 - Similarity matrix implementation

### 6.6.6 Creative text templates identification for creative title

In creative title generation finding templates is important task. For these sentences can be passed in vast size hence the identification task should be minimal. Hence without heavy NLP work with simple similarity-based work implemented.

In templated identification each template match against title with cosine similarity. Checking cosine similarity between title and template vector directly doesn't give accurate results all the time hence, as an alternative cosine similarity between keywords of title and template also considered. Similarity for each keyword in template added similarity check against keyword in title and an average value calculated. A template is added to candidates of the sentence similarity and keywords similarity greater than 0.5. Finally, templates sorted based on average similarity score.

```

def search_candidates_for_creative_sentences(self, title, templates):
    candidate_templates = []
    for template, _ in self.nlp.pars_document(templates, as_tuples=True):
        template = Template(template)
        template.nlp_text = self.nlp.pars_sentence(clean_sentence(template.doc.text))

        sentence_similarity_score = self.title_sentence_similarity(title, template)
        keyword_score = self.keyword_score(title, template)

        if sentence_similarity_score >= 0.5 and keyword_score >= 0.5:
            candidate_templates.append(((sentence_similarity_score + keyword_score) / 2, template))

    candidate_templates = sorted(candidate_templates, key=lambda tup: tup[0], reverse=True)
    candidate_templates = [candidate for _, candidate in candidate_templates]
    return candidate_templates

```

Figure 6.17 - Search candidate implementation

```

def title_sentence_similarity(self, title, sentence):
    similarity_score = 0
    if title.doc.has_vector and sentence.nlp_text.has_vector:
        similarity_score = title.doc.similarity(sentence.nlp_text)
    return similarity_score

```

Figure 6.18 - Title sentence similarity implementation

```

def keyword_score(self, title, sentence):
    i = 0
    token_score = 0
    for index, similar_words in title.important_keyword_similar_words.items():
        for similar_word in similar_words:
            for token in sentence.nlp_text:
                if not is_valid(token):
                    continue

                try:
                    vocab_id = self.nlp.vocab.strings[similar_word]
                    vocab_vector = self.nlp.vocab.vectors[vocab_id]
                    s = cosine_similarity([token.vector], [vocab_vector])[0][0]
                    if s > 0.5:
                        token_score += s
                        i += 1
                except KeyError:
                    pass

    return token_score / i if token_score != 0 else 0

```

Figure 6.19 - Keyword score implementation

### 6.6.6.1 Selectional Preference Strength Calculation

SPS of a verb is calculated using word selectional preference data and the verb. The formula of SPS calculation explained in [Section 3.5](#). First it retrieves the noun sum, verb total and noun total from selectional preference data. The initial SPS value set as 0 in case no noun found for verb in selectional preference data. Then prior probability and posterior probability calculated. Finally, SPS calculated by summing it up.

```

def sps(self, verb, noun_totals=None, noun_sum=None):
    """Calculates and returns SPS(verb)"""
    try: # If it never occurs in the corpus, return None
        self.d[verb]
    except KeyError:
        return None

    SPS = 0
    if not noun_sum:
        noun_sum = self.noun_sum()
    verb_total = self.verb_total(verb)
    if not noun_totals:
        noun_totals = self.noun_totals()
    for noun in self.d[verb].keys():
        if noun:
            prior_prob = self.noun_prob(noun, noun_totals, noun_sum)
            post_prob = self.conditional_prob(verb, noun, verb_total=verb_total)
            SPS += post_prob * math.log(post_prob / prior_prob)

    return SPS

```

Figure 6.20 - Selectional preference strength calculation implementation

```

def noun_prob(self, noun, noun_totals, noun_sum):
    """Returns P(noun)"""
    return noun_totals[noun] / noun_sum

```

Figure 6.21 - Noun probability calculation



```
def conditional_prob(self, verb, noun, verb_total=None):
    """Returns P(noun|verb)"""
    if not verb_total: # For efficiency
        verb_total = self.verb_total(verb)
    try:
        return self.count(verb, noun) / verb_total
    except ZeroDivisionError:
        return 0
```

Figure 6.22 – Conditional probability calculation

### 6.6.6.2 Selectional Association Calculation

Selectional Association of verb and noun calculated using SPS value and selectional preference data. SA formula explained in [Section 3.5](#).

```
def sa(self, verb, noun, noun_totals=None, noun_sum=None, sps=None):
    """Calculates and returns SA(verb)"""
    try: # If it never occurs in the corpus, return None
        self.d[verb][noun]
    except KeyError:
        return None
    if not noun_sum:
        noun_sum = self.noun_sum()
    if not noun_totals:
        noun_totals = self.noun_totals()
    if not sps:
        sps = self.sps(verb, noun_totals=noun_totals, noun_sum=noun_sum)

    SA = 1 / sps
    SA *= self.conditional_prob(verb, noun)
    SA *= math.log(self.conditional_prob(verb, noun) / self.noun_prob(noun, noun_totals, noun_sum))
    return SA
```

Figure 6.23 - Selectional Association calculation

### 6.6.7 Find important words in title and templates

As discussed in [Section 6.4.3](#) three types of word concepts derived for important words. Named entities and word with more probability appear in titles are derived. Important words and named entities deriving for sentence code in Figure 6.24. The third type of word concept similar words for keywords derived from Word2vec embeddings mostly synonyms and related are returned figure for similar words in Figure 6.25.

```

def important_keywords_indexes(self, tokens):
    i = -1
    keyword_index = []
    for token in tokens:
        i += 1
        try:
            if not is_valid(token):
                continue
            # Checking named entity
            if token.ent_type != 0:
                keyword_index.append(i)
                continue

            # Checking word is an important word for words collected from corpus
            word = token.text if token.pos in [PRON, ADJ] else token.lemma_
            val = self.word_coverage[word.lower()]
            score = val / headline_count
            if score > keyword_threshold:
                keyword_index.append(i)
        except KeyError:
            pass

    return keyword_index

```

Figure 6.24 - Important keyword implementation

```

def substitute_words(self, w, POS, sentence):
    """
    Get appropriate substitution for a word given context words
    """

    # generate candidate substitutions
    candidates = self.get_candidates(w, POS)
    if sentence is None:
        return candidates[:self.n_substitutes]
    else:
        context_words = tools.get_words(sentence)
        # filter context words: exist in the word2vec vocab, not stop words
        context_words = list(filter(lambda c: c in self.word_vectors.vocab
                                   and c not in tools.stopwords,
                                   context_words))
        cand_scores = [self.get_substitutability(w, s, context_words) if s in self.word_vectors.vocab else 0 for s
                       in candidates]
        assert (len(cand_scores) == len(candidates))
        sorted_candidates = sorted(zip(candidates, cand_scores), key=lambda x: x[1], reverse=True)
        return [sub for sub, score in sorted_candidates][:self.n_substitutes]

```

Figure 6.25 - Substitute words implementation

### 6.6.8 Generating creative title

Generating creative title consists of number of subtasks. The title and templates go through the subtasks to generate creative title. First candidate templates are identified. The enhance title is the process identifying important words for title.

```

def generate(self, titles, templates):
    candidates = []
    for title in titles:
        title = self.enhance_title_info(title)
        creative_sentences = self.search_candidates_for_creative_sentences(title, templates)
        for creative_sentence in creative_sentences:
            v, replaced, inserted = self.substitute_words(creative_sentence, title)
            if replaced or inserted:
                candidates.append(creative_sentence.text)
                print(
                    f"template:{creative_sentence.text} | modified:{v} | replaced:{replaced} | inserted:{inserted}")
    return candidates

```

Figure 6.26 - Creative title generation implementation

In the process of substituting words for templates, suitable replacement word for title and template. Replacement words should match one of the criteria of same part of the speech (POS), adjective or adverb with suitable similarity score or suitable named entity with suitable alias or pseudonym name.

```

def substitute_words(self, temple_sentence, title):
    important_keywords = title.important_keyword_indexes
    replace_words = {}
    insertion_words = {}

    for index in important_keywords:
        title_token = title[index]
        if not is_valid(title_token):
            continue
        i = 0
        for token in temple_sentence.doc:
            if not is_valid(token):
                i += 1
                continue
            # Replace word
            if token.pos is title_token.pos:
                score = token.similarity(title_token)
                if score > 0.5 and score != 1:
                    replace_words[i] = title_token
            # Insert word as adjective
            elif i > 0 and token.pos == NOUN and title_token.pos == ADJ:
                score = token.similarity(title_token)
                if score > 0.5 and score != 1:
                    insertion_words[i] = title_token
            i += 1

    sent = [i for i in temple_sentence.doc]
    for index, val in replace_words.items():
        sent[index] = val

    for index, val in insertion_words.items():
        sent[index - 1] = val

    return " ".join(sent.text for sent in sent), len(replace_words) > 0, len(insertion_words) > 0

```

Figure 6.27 - Substitute words implementation

## 6.7 Problems Encountered

During the development time several problems encountered related to framework and data gathering.

- Creative title generation and identification task requires amount resource (RAM) at runtime to load hence less memory is available for text processing so the rest of the memory should be optimized. Considering this problem every task implemented considering memory optimization.
- In identifying suitable templates for title not always selected suitable template because of the vectors of a sentence is average of vectors of words in the sentence hence it won't accurate. To overcome this issue a keyword similarity checked also implemented. Both title and template keywords derived and a sum of score calculated. Then condition changed to, if title and template sentence similarity and keywords similarity greater than threshold value it considered as candidate.
- In vectorizing words process, word2vec doesn't contain every possible this cause problem in accuracy of identification hence, the most similar word

## 6.8 Chapter summary

A critical evaluation of tools and technologies was done and then the tools and technologies most appropriate for this project were selected. It was decided to use spaCy mainly because of the fastest and most accurate parsing feature. It was decided to use sci-kit learn for clustering because of its better and fast clustering results. As mentioned in the chapter, appropriate algorithms were applied for developing each feature. There were some issues during the implementation, but it was possible to overcome those issues.

# **Chapter 7: Testing**

## 7. Testing

### 7.1 Chapter Overview

This chapter discusses the testing phase of the project. The testing goals are defined, and the testing methods are selected to achieve the set goals. The chapter discusses about the different testing methods and testing tools that are chosen based on the project requirement. The chapter also presents the test cases with their results. Furthermore, the chapter analyzes the test results and provides remarks about the results. Finally, the testing process was reviewed, and the pros and cons of the process were outlined.

### 7.2 Testing Goals

Testing is useful in order to guarantee the framework is bug-free and meet the requirements. Testing goals represents whole testing tasks motive. Following are the main goals of testing.

- Recognize bugs expected and unexpected bugs and alternate framework to make sure the final version contains minimum set of defects and errors.
- Prove functional and non-functional requirements of the framework are met.
- Improve and optimize framework based on the test results.

### 7.3 Test application description

In order to test the framework an end application required hence an application trained on creative text identification from corpus and dataset then store the creative text in a database. Number of news and articles collected from online. The goal of the application is to generate creative title for news from creative text stored in database and title of the news. Test corpus and creative text information discussed in upcoming sections.

### 7.4 Testing criteria

There are many software testing methods currently available. Following is a brief description of some of the major testing methods used to test the framework.

#### 7.4.1 Functional Testing

##### Unit testing

The framework has number of sub tasks in each module and some tasks and a module used a helper module this makes the framework fairly complex. Therefore, it is essential to conduct test every units of the framework before incorporating every units with complex modules. Also, it is more useful when instantly distinguish bugs in units its make development quicker without wasting time later. Furthermore, it enhances the code quality and diminish mistakes, since it

checks each conceivable way a technique can work. Based on these reasons' unit test chosen for the framework.

### **Module and Integration Testing**

This technique could be used test individual modules as well as integrated modules in the framework to make sure they function as expected. Since the modules are communicate each other in when it extended. Integration test is also required width module testing as some modules depends on other modules to perform the functionality. Therefore, it was selected as the testing methodologies in this project.

## **7.4.2 Non-Functional Testing**

### **Accuracy Testing**

Accuracy identified as critical non-functional requirement for the framework. The effectiveness identification and generation modules highly depend on the accuracy. Based on these reasons' accuracy testing used for the framework.

### **Performance Testing**

A non-functional testing used determine stability and responsiveness of a framework under various work load situation. This could be used to determine the resource usage of the framework and determine the time taken to complete the cycle of adaptation which are critical to be tested in order to provide the best user experience to the user.

## **7.5 Selection of Testing Framework**

There are wide range of testing frameworks available for unit testing for python such as PyTest, unittest and Nose. Among the PyTest is the popular industrial standard framework for testing. It offers exceptionally progressed and modern highlights for testing. PyTest identified as suitable testing framework based on following features.

- Allows for minimized test cases
- Ease of use
- It serves to parameterize any test and cover all references of a unit without code duplication
- The configuration is straightforward and convenience

## **7.6 Test execution and results**

Testing for functional and non-functional test method process will be discussed with test cases process and results.

## 7.6.1 Unit Testing

Execution of unit testing explain in following table and for detailed test execution and results check [Appendix F](#). For some functional requirement several test cases created because it contains multiple subtasks.

FR-ID	No. of test cases	Passed test cases	Final Status	Pass Rate %
FR1	2	2	Pass	100
FR2	4	4	Pass	100
FR3	10	10	Pass	100
FR4	3	2	Pass	67
FR5	4	4	Pass	100
FR6	1	1	Pass	100
FR7	1	1	Pass	100
FR8	3	3	Pass	100
FR9	1	1	Pass	100

*Table 7.1 - Unit test results summary*

Examining the unit test result table, it can be seen that all test case identified with functional requirements are passed except non-English language test case for generating title. It is not considered as critical for initial prototype due to limited time for the project. It will be implemented for future release.

## 7.6.2 Module and Integrating Testing

Module test consist of different units and all the units are test together. Different types of module and integration testing available among them Big Bang approach and Incremental approach are selected for review. Big Bang approach integrate everything together and test at once. This approach is easy to implement and also comes with drawbacks identifying correct place where the test case failing it difficult and since all modules are test together important modules which should be tested in isolate may cause error later. Even though this module more suitable for small scaled project considering the drawbacks it identified as not suitable. Incremental approach is more convenient for the project because it tested by joining modules which are logically related. This test goes on until all the modules are joined and



tested. In this way bugs in module may revealed and finding where it the bug is occurs is easy. Based on the review incremental approach selected for module and integrating test. Incremental approach has three divisions,

- Top Down Approach
- Bottom Up Approach
- Sandwich / Hybrid Approach

The bottom up approach is selected because of following reasons.

- Bug identification is easy
- Not to wait until all modules are implemented
- Test cases are simple to make

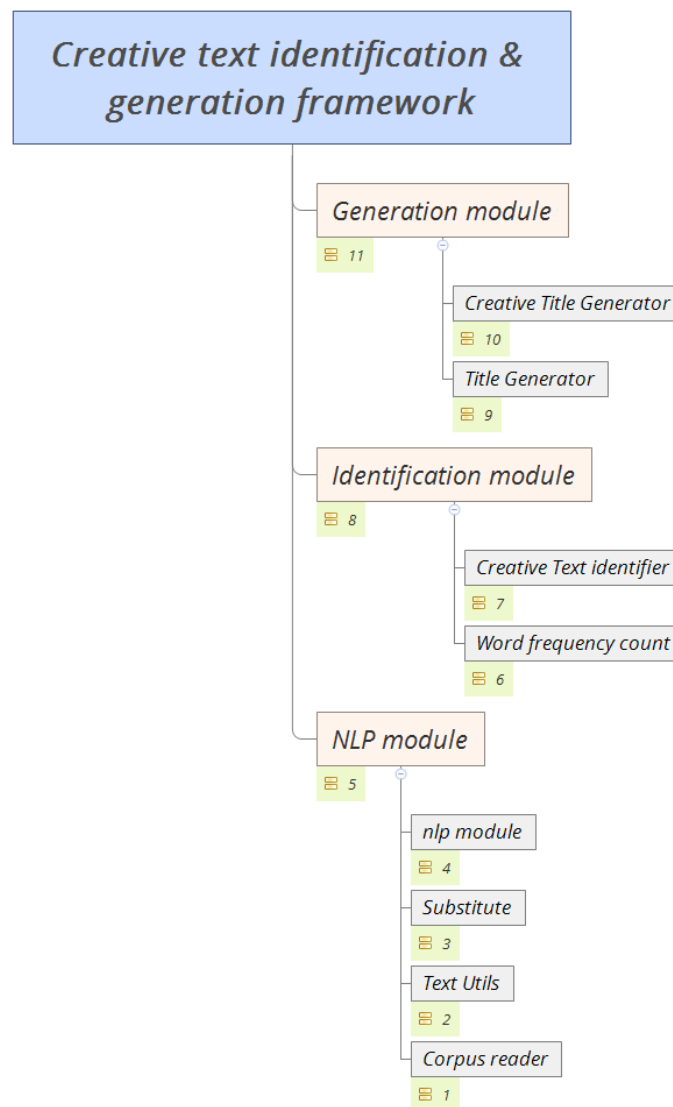


Figure 7.1 - Framework testing modules

Bottom up integration testing will be followed as shown in Figure 7.1. Summary of module integration test showed below and for detailed test results check [Appendix G](#).

#	Module	Test Case	Result
1	NLP Module	Read contents from corpus reader	Pass
2	NLP Module	Identify substitute words in sentence	Pass
3	NLP Module	Parse sentence	Pass
4	Identifier module	SA calculation on word frequency	Pass
5	Identifier module, NLP module	Creative text identification from raw sentence list	Pass
6	Identifier module, NLP module	Creative text identification from corpus reader object	Pass
7	Generator module	Generate title from sentences	Pass
8	Generator module, NLP module	Generate title from corpus reader object	Pass
9	Generator module, NLP module	Creative title generation	Pass

*Table 7.2 - Module integration test results*

Modules tested with collaborating with other modules and based on Table 7.2 every test case is passed and modules are working fine.

### 7.6.3 Accuracy Testing

Following components distinguished to test accuracy on test application

- Creative text identification
- Find keywords for sentence
- Finding suitable templates for creative title generation

Creative title generation wasn't tested because there is no way statistical measure for creativity of text generation

### 7.6.3.1 Accuracy testing for creative text identification

Creative text identification is one of the crucial features of the framework. Creative text identification should be able to identify different types of creative text among them metaphor is an important creative text also creative text identification's main concern is to identify creative text by identifying the conceptual blending in the sentence and metaphor represents conceptual blending more than other creative text hence, accuracy of the creative text identification tested on VU Amsterdam Metaphor corpus (VUAMC) (Steen *et al.*, 2010) a metaphor annotated corpus. To extend the creative text types a small data of clichés and expressions also included.

Preparation for testing done as following. The VU Amsterdam Metaphor corpus is preprocessed and by extracting all raw text and marking each metaphor relation word with "*functions = mrw*". Then a script executed on the corpus and converted the corpus into csv format with marking metaphor usage words with "*M\_*". From Amsterdam corpus, clichés and expression dataset sentences derives. Using spaCy parser 40,622 verbs, of which 23,069 have at least one subject or object relation were identified from sentences.

Sentences were extracted and formed into two types of dataset 1. Sentences annotated with metaphors 2. Sentences without annotation. Sentences without annotation dataset passed creative text identification and compare against annotated corpus. Formula for accuracy test.

$$Accuracy = \frac{\text{Number of verb and (object or subject) pair in identified sentences}}{\text{Number of verb and (object or subject) pair in annotated sentences}} * 100$$

*Equation 7.1 – Creative text identification accuracy test*

#### Verb with only Subject pairs

$$\frac{10,938}{13,466} * 100 = 81.23 \%$$

#### Verb with only Object pairs

$$\frac{3,269}{3,913} * 100 = 83.54 \%$$

#### Verbs with both a Subject and an Object triples

$$\frac{4,320}{5,539} * 100 = 77.99 \%$$

## Average Accuracy

$$\frac{\text{Identified (verb with subject + verb with object + verb with subject and object)}}{\text{Annotated (verb with subject + verb with object + verb with subject and object)}} * 100 = \text{Accuracy}$$

$$\frac{10,938 + 3,269 + 4,320}{13,466 + 3,913 + 5,539} * 100 = 80.84 \%$$

Throughout the evaluation, the accuracy of creative text identification identified as 80.84 %. The accuracy dropped in verb with both a Subject and an Object triples because verb has both subject and object hence, the task has to identify both correctly as creative to pass the evaluation.

### 7.6.3.2 Finding keywords for sentence

Finding keywords for sentence is crucial task in creative title generation. For testing dataset of news title collected for end application was used. The part of dataset formed with sentences and keywords picked by individuals. Equation of accuracy of finding keywords for each sentence as follows

$$\text{Accuracy} = \frac{(\text{same keywords of framework} - \text{additional keyword})}{\text{keywords of individuals}} * 100$$

*Equation 7.2 - Accuracy of finding suitable keywords*

Summary of results in table detailed results on [Appendix J](#)

#	Sentence	Summed keywords	Accuracy (%)
1	Santos says mud disaster funds appropriate	1	33
2	firefighters join Vic fire effort	2	100
3	cleaners march through cbd over pay conditions	2	67
4	hurricane Ivan kills 10 in Caribbean	3	100
5	tonnes of oil blanket Queensland beaches	2	67
6	large two storey factory engulfed flames at Williamstown north	0	0
7	rain brings welcome relief for firefighters	1	33

8	high winds ground balloon championships	1	50
9	costello defends future fund move	4	100
10	children injured in train ride accident	3	100

Table 7.3 - Summary results of finding keywords

Based on Table 7.3 average calculated

$$\frac{(33 + 100 + 67 + 100 + 67 + 0 + 33 + 50 + 100 + 100)}{10} = \frac{650}{100} = 65\%$$

The accuracy for finding suitable keywords for sentence is 65%. Accuracy of keyword identification lower than expected but the model is capable enough identify at least two keywords in sentence due additional keywords finding accuracy level dropped. Since the model extract named entities accuracy increases a bit. Other than this additional finding issue keyword finding work as expected.

### 7.6.3.3 Accuracy testing on finding suitable templates

To test finding suitable templates for title, five more suitable title and template selected manually, and twenty more random templates selected. Then, each title and more suitable template with other random templates given to framework. The accuracy of finding more suitable template for title is calculated based on the ranking position of more suitable template for title with other random templates.

Formula of a test case is based on highest rank is more accurate.

$$Accuracy = \frac{(number\ of\ elements + 1) - rank\ of\ identified\ template}{number\ of\ elements} * 100$$

Equation 7.3 - Finding suitable template accuracy

And average value of test cases calculated as overall accuracy of finding suitable template. Suitable title and templates pair can be found in [Appendix H](#) and random templates in [Appendix I](#).

Result of accuracy testing for suitable templates

#	Title	Template	Rank of identified template	Accuracy (%)
1	The Obama administration is planning to issue a final rule designed to enhance the safety of offshore oil drilling equipment.	Bridge over troubled water	1	100
2	Russia's defense ministry has rejected complaints by U.S. officials who claimed Russian attack planes buzzed dangerously close to a U.S. Navy destroyer[...]	The empire strikes back	3	90
3	There will be no soft Brexit now. It's no deal, revoke or another vote	Throw cold water on	3	90
4	Doctor describes 'ecstatic' moment coma patient woke up after 27 years	He's waiting for his ship to come in	5	80
5	Eden Hazard snubbed for Player of the Year	Turn a blind eye	1	100

Table 7.4 - Results of test finding suitable template

Based on table 7.3 the average accuracy of test is

$$\frac{(100 + 90 + 90 + 80 + 100)}{5} = \frac{460}{5} = 92\%$$

The accuracy finding suitable template calculated as 92%. The reason behind the fourth title, template ranked 5 because of keyword scoring was lower than expected common similar words found very less.

#### 7.6.4 Performance Testing

Following are the recognized components in-order to calculate the performance of the core functionality. More insights about the core functionality of the framework.

- Creative text identification
- Creative title generation

Every test case executes five times and average value is taken. Following equation followed for test cases.

$$Performance = \frac{\sum(\text{execution time for each time})}{\text{Number of attempts}}$$

Equation 7.4 - Performance test case

## Testing Environment

Test cases are done on a machine with Core i7-8750H CPU, 8 GB RAM and Windows 10 OS.

### 7.6.4.1 Creative title generation

A performance testing of the creative title generation was completed to test the performance time of the framework. Tested for a title and 100 template sentences.

#	Test case	Input	Expected time	Actual time	Results	Remarks
1	Loading libraries and models	N/A	1.2 minute	1.13 minute	Pass	Typical
2	Searching templates	Creative sentences list	1.5 minute	1.32 minute	Pass	Typical
3	Extracting and expanding keywords	Set of words	2 seconds	1.1 second	Pass	Typical
4	Lexical substitution	Title and keywords	2 seconds	1.1 second	Pass	Typical

Table 7.5 - Creative title generation performance testing

Loading libraries and models, searching templates task work as expected but both takes too much time. Since framework modules depend on several models it takes time to load and searching templates goes to loop with comparison. Other than these two issues creative title generation performance is better.

### 7.6.4.2 Creative text identification

Performance of creative text identification decided by number sentence it process. Number sentences given to identification task multiplied two each execution. The identification task's performance depends on spaCy and sci-kit learn libraries.

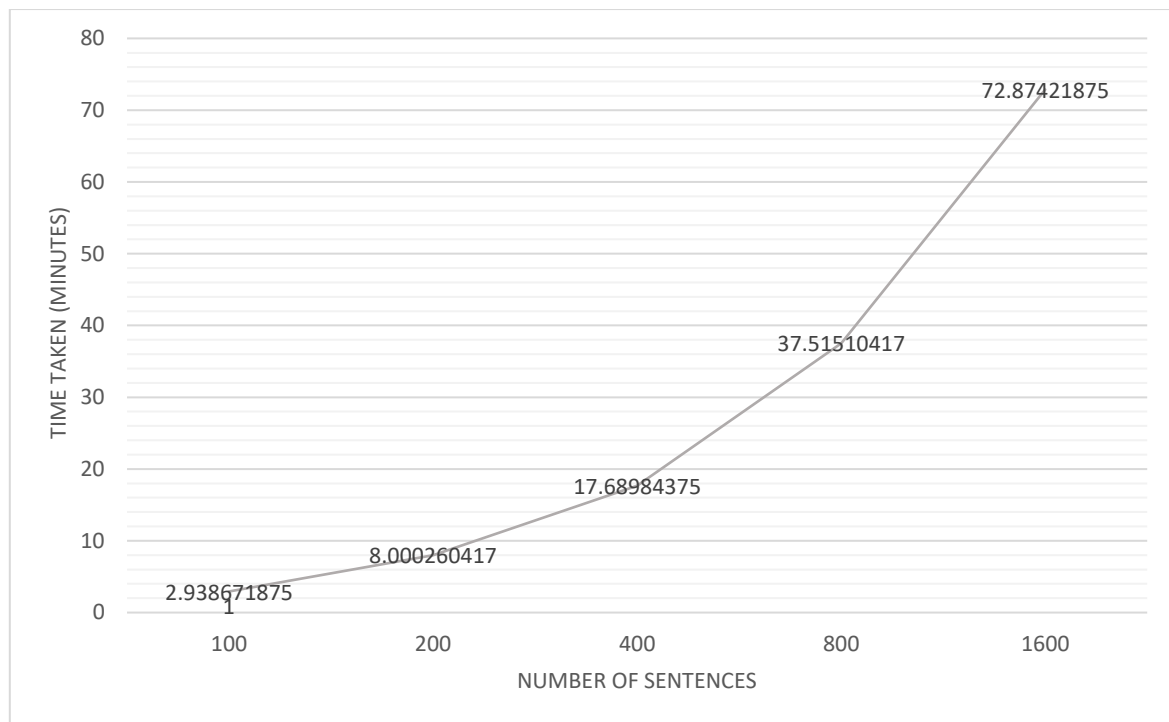


Figure 7.2 - Creative text identification performance

Based on Figure 7.2, time complexity of creative identification is  $O(n^2)$ . The identification task's time increases in quadratic manner. The main reason behind the quadratic increment is parsing taken by spaCy other than parsing clustering, vectorizing, SPS and SA calculation perform in a linear manner this means the module perform well as expected.

## 7.7 Remarks and Limitation of testing

There was some limitation identified during testing phases some of them discussed here.

### Lack of human judgment

The project builds upon on main factor creativity. Measuring creative still a struggling matter (Said-Metwaly, Noortgate and Kyndt, 2018) it cannot be measure only by machines. It requires more human knowledge & judgment. Due to limited time testing with humans was not carried out for creative text identification and creative title generation features which is a huge downside in the testing process.

### Data set and domain limitation

Creative text identification should be tested different types of creative text and domains due limited amount time and data set. The identification task tested only expressions and metaphor data sets in news domain. The accuracy test of other creative text type not carried due to limited time and testing every type isn't possible.



## 7.8 Chapter Summary

This chapter focused on testing the project and documenting the test results. First the testing goals were defined, and the testing methods were selected to achieve the goals. It was decided to perform both functional and non-functional testing to verify and validate the requirements of the framework. Unit testing and module and integrating testing were carried out to test the functional requirements. The testing results confirmed that the high priority functions are working properly. Accuracy Testing and performance testing were carried out to test non-functional requirements. The testing results showed a good level. The chapter also discussed about the testing tools used in the process. Thereafter, the pros and cons of the testing process were discussed.

## **Chapter 8: Evaluation**

## 8. Evaluation

### 8.1 Chapter Overview

This chapter focuses on evaluating the framework at different aspects such as concept, usability and technical aspects. The evaluation goals are set based on the main areas that needs to be assessed. Thereafter, the evaluation criteria are decided based on the evaluation goals. Identified aspects are evaluated either quantitatively or qualitatively and the result is discussed. Furthermore, the chapter includes benchmarking of the framework by comparing it with other online indexing systems. Finally, a critical evaluation is presented, and the pros and cons of the system are discussed in depth.

### 8.2 Evaluation Goals

The evaluation objectives are set to evaluate the main aspects of the framework. Following areas identified as main goals of the project.

#### 8.2.1 Evaluation of the concepts

Even though existing frameworks provide solution at some level, it is important to prove that the framework and approaches used by the project provide better solution for the problem in better way. Goals of concept evaluation given below.

- Scope and depth of the project
- Need for creative text identification and generation framework
- Use and impact of the framework in the field.

#### 8.2.2 Evaluation of the Technical Aspects

As a software engineering project, it is vital to evaluate the technical aspects of the framework. The end goal to check whether the suitable coding benchmarks are achieved, and the framework meets the industry standard.

- Framework Architecture and Design
- Coding practice
- Tools and technologies
- Developer friendliness of the framework

#### 8.2.3 Evaluation of the Usefulness and Impact of framework

Valuation of the usefulness and impact is important to identify whether the framework solves in developer's problem in effective way. Besides, it emphasizes how valuable the result of the

project and the productivity of innovative use of the framework compared with the other similar systems. Goals of the usefulness and impact given below.

- Features of the framework
- Accuracy and performance of framework.

## 8.3 Selection of Evaluators

### Software Engineering Experts

SE experts such as Technical Leads were chosen to get feedback on framework design, creativeness of generation and identification accuracy.

### Marketing Industry Experts

Marketing is a field where creativity is considered as a main aspect in the field hence experts on marketing field chosen to get criticism about the concepts, usefulness and future enhancements of framework.

Evaluators qualification and designations can be found in [Appendix E](#)

## 8.4 Evaluation Methodology

A prototype application developed on top of the framework for evaluators to identify the use of the framework and make the evaluation easier. The project was evaluated using interviews, questionnaires and statistical analysis. Interviews were conducted to evaluate the qualitative and quantitative aspects of the prototype application. Questionnaires directed at the interviews were prepared to measure the following.

- Qualitative measures of the overall concept and scope of the framework, architecture, design and implementation of the system, prototype and the algorithms.
- Quantitative measures of the non-functional requirements of the system.

Other than interviews and questionnaires a statistical evaluation followed to evaluate the accuracy of creative text identifier part of the framework. VU Amsterdam Metaphor corpus (VUAMC) (Steen *et al.*, 2010) will be used to evaluate the part. Creative text identification should be able to identify different types of creative text and metaphor is among them and the main focus of creative.

## 8.5 Execution of evaluation methods

The evaluation feedback categories as discussed in [Section 7.2](#). Both qualitative and quantitative evaluation available.

## 8.5.1 Evaluation of the concepts

### 8.5.1.1 Scope and depth of the project

<b>Question</b>	What is your thought on scope and depth of the project?
<b>Feedback</b>	<p>“The scope and the depth of the application looks sufficient, it provides good set of features for catchy title generation with discovering creative text and phrase article. Also, the features provide detailed information”</p> <p>“The scope is defined well”.</p>
<b>Review</b>	Based on feedback defined scope for the project meet the expected depth. NLG and NLU are new area and its broad area to study. This project involves both areas so, the scope and depth are acceptable and sufficient enough for an undergraduate student as per the feedback received.

*Table 8.1 - Scope and depth evaluation*

### 8.5.1.2 Need for creative text identification and generation framework

<b>Question</b>	What is your general idea about the framework need?
<b>Feedback</b>	<p>“It is valuable to have framework like this, now in industry like marketing innovation is key component. People tend come up with new ideas with different way this could fill the gap and hep people to think better.”</p> <p>“This framework is capable enough to automate the process of generate and identifying creative text with few enhancements we could see it as industrial material.”</p>
<b>Review</b>	As evaluators reviews, they say this the framework is valuable, and this framework can be used in areas where creativeness and innovation are considers as main components because it brings new ideas. Also, evaluators believe this framework can automate the work on their own and with few enhancements this framework could be industrial product. Based on the review with the work of future enhancements this framework could be valuable product, so the proposed solution is appreciated and has great potential on future.

*Table 8.2 - Need for creative identification evaluation*

### 8.5.1.3 Use and impact of the framework in the field

Question	Provide your thought how useful is the framework in your field?
<b>Feedback</b>	<p>“Creativity is the backbone of advertising. When advertising products and services advertisers have to come up with catchy phrases relative to consumer’s area. This product would fill this area”</p> <p>“I believe this application is valuable to come up with new ideas for products and services. It will be useful for advertising and media industry.”</p>
<b>Review</b>	<p>Based on feedback evaluators believe this framework more useful where creativity and new ideas are mainly conserved areas such as marketing, advertising and media. Evaluators satisfied with framework and believes it useful.</p>

*Table 8.3 - Use and impact in the field evaluation*

## 8.5.2 Evaluation of Technical Aspects

### 8.5.2.1 Framework Architecture and Design

Question	What is your opinion about modularizing components of framework?
<b>Feedbacks</b>	<p>“modularising classes is an effective way of losing components depending on each other. Here the identification, generation and NLP modules are separated this provides a purpose for developers.”</p> <p>“Works are divided among modules and management is easy”</p> <p>“Modular architecture forces to decouple components and increases the usability”</p>
<b>Review</b>	<p>Decision selecting modular architecture in the framework welcomed by evaluators. They believe that modularizing components prevents depending each other and managing components will be easier. Based on the feedback choosing modular architecture is accepted by evaluators.</p>

<b>Question</b>	What is your opinion about pipeline architecture of creative text identification?
<b>Feedbacks</b>	<p>“Choosing pipeline architecture is good move. Identification process goes through series and the object are modified and used in another pipe. Well thought out plan!”</p> <p>“Effective choice. It will increase the performance. “</p> <p>“Pipeline architecture increases the processing speed the system. Identification has to process lot of text hence pipeline architecture is good choice. There are also some drawbacks when customizing components. it makes system complex”</p>
<b>Review</b>	<p>Evaluators believe that pipeline architecture makes huge impact on performance of creative identification task. The given sentence is modified in different objects in each task of series of tasks so, it is suitable. Also, they said that when extending, pipeline architecture would make extending framework difficult when complicated logics to be implemented. This drawback won't be problem in the framework because it provides required methods to be extended even it more than required developer can customized identification task itself this gives more freedom in customizing. Other than drawback evaluators appreciated the choice.</p>

*Table 8.4 - Framework architecture and design evaluation*

### 8.5.2.2 Coding practice

<b>Question</b>	What is your opinion about coding practices and implementation of the framework?
<b>Feedbacks</b>	<p>“Following style guideline pep 8, detailed naming convention and OOP concepts are appreciated”</p> <p>“Best practices are followed. Implementation requires more resources on run time choosing light weigh packages more effective”</p> <p>“Python style guide followed, and module are implemented in standard way. A part of framework title generation takes too much RAM power it should be concerned”</p>

<b>Review</b>	Based on feedbacks from evaluators. They apprentice the coding practices followed in the framework. Some complain about high RAM usage on runtime. The project's non-functional requirements are accuracy and performance for the prototype. This issue will be considered in future releases.
---------------	--

*Table 8.5 – Coding practices evaluation*

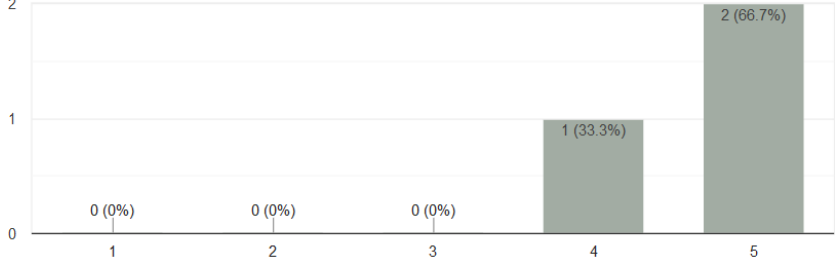
### 8.5.2.3 Tools and technologies

<b>Question</b>	What is your opinion on selecting spaCy for parsing sentence framework?
<b>Feedbacks</b>	<p>“Spacy is the fastest available for parsing. So, the performance of the framework would be better.”</p> <p>“Good choice because spacy has better accuracy. But it introduced recently and it's a research project Core NLP would have better choice”</p> <p>“Spacy's accuracy and performance are better the CoreNLP library for parsing. Also Spacy is industrial standard library. well choice”</p>
<b>Review</b>	spaCy is the fastest and most accurate library available for parsing. Hence, evaluators believe this increases the performance of the framework. Evaluator said spaCy's immaturity would affect the project. Even though spaCy released recently it adapted tech giant companies. Also, the documentation of spaCy is well defined. This won't affect the framework. spaCy's compared with CoreNLP and they said spacy is better. Based on the feedback given by evaluators choosing spaCy for parsing is a well choice.

*Table 8.6 - Tools and technologies evaluation*



### 8.5.2.4 Developer friendliness of the framework

<p><b>Feedbacks</b></p>	<p>How would you rate this framework setup for a end application? based on steps to install library, download models and extending framework</p> <p>3 responses</p>  <p style="text-align: center;"><i>Figure 8.1 - Developer friendliness rating evaluation</i></p> <p>Please give me the reason for above rating</p> <p>3 responses</p> <ul style="list-style-type: none"> <li>Integrate framework with pip library is a good move. It make developer install package in easy way</li> <li>Easy setup with few commands</li> <li>Download model separately and implemented in pip package</li> </ul> <p style="text-align: center;"><i>Figure 8.2 - Developer friendliness rating reason evaluation</i></p>
<p><b>Review</b></p>	<p>Based on Figure 8.13 evaluators rated four and five as average rating 4.7 for setting up the framework and they reason provides the framework developed in pip library it is easy to install and maintain. Moreover, the models, such as word frequency, cluster and important keywords are downloaded separately with a command. Overall evaluators satisfied with framework setup.</p> <p style="text-align: center;"><i>Table 8.7 - Developer friendliness evaluation</i></p>

### 8.5.3 Evaluation of Usefulness and Impact

#### 8.5.3.1 Features of the framework

<p><b>Question</b></p>	<p>What is your opinion about using selectional preference method (statistical method) on creative text identification of framework?</p>
<p><b>Feedback</b></p>	<p>“Using a statistical based method is good choice for creativity-based project because identifying creativity is difficult thing and statistical methods has</p>

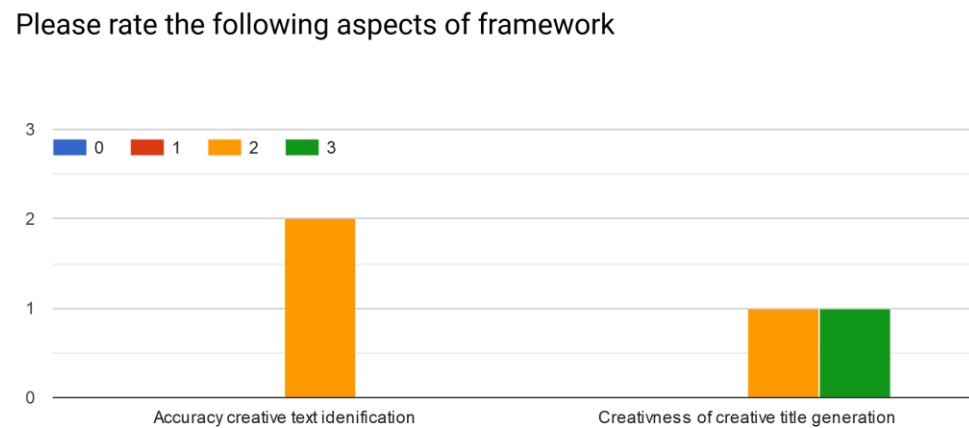
	proven identifying certain level of creativeness in text. Also, this method could extend to different language other English”.
<b>Review</b>	Evaluators satisfied with selectional preference method because it’s a statistical based method so it’s more suitable for discover creativeness in text.
<b>Question</b>	What is your opinion about using lexical substitution on creative title generation of framework?
<b>Feedback</b>	“Lexical substitution method prevents grammatical error in sentence generation. Using lexical substitution, the result of process in known or could be guessed hence the method trustworthy”.
<b>Review</b>	Choosing lexical substation method welcomed by evaluator because less grammatical error and can predict the outcome

Table 8.8 - Features of framework evaluation

### 8.5.3.2 Accuracy and performance of framework

**Feedback**

Please rate the following aspects of framework



Aspect	Rating 0	Rating 1	Rating 2	Rating 3
Accuracy creative text identification	0	0	2	0
Creativness of creative title generation	0	0	1	1

*Figure 8.3 - Prototype aspects evaluation*

What do you think about the aspects of framework ?

2 responses

The application able to identify catchy word & phrases in the article and generated title also catchy and relative for article

Different types creative words identified and words correctly applied to created title

*Figure 8.4 - Aspects of framework evaluation*

<b>Review</b>	From Figure 8.3 evaluators voted 2 and 3 rating for the accuracy of text identification and creativeness of title generation based on this vote, evaluators satisfied with the framework aspect's accurate and creativeness. As shown in Figure 8.4 evaluators believes the creative text identification's coverage of creative word type is wide. Also identified words are relative which means it is reusable for generation task. Then the framework substitute without any fault and substitution words are relative for title.
---------------	--

Table 8.9 - Accuracy of framework evaluation

## 8.6 Benchmarking

Benchmarking was done to evaluate the project with existing solution. The similar solutions are described in literature review [Section 3.4](#). Project compared the features of existing frameworks.

	PersuAIDE! (Munigala <i>et al.</i> , 2018)	Slogans are not forever (Gatti <i>et al.</i> , 2015)	BRAINSUP (Özbal, Pighin and Strapparava, 2013)	CreativeAid!
Force words to appear in generated sentence	✓	X	X	X
Generate creative sentence from keywords	✓	X	X	X
Generate sentence from creative sentence	X	✓	✓	✓
Generate sentence from document	X	X	X	✓

Identify creative sentences for generate	X	X	X	✓
Read from corpus	X	X	X	✓

Table 8.10 - Comparison of features

Based on Table 8.1, it can be concluded that this framework has most of the features. Still some features are missing from the framework due to limited time those features aren't implemented. These features will be considered in future releases.

## 8.7 Research Questions and Answers

The following are the main three research questions for the project. Questions identified in [Section 1.6](#)

- a. What is the best approach to identify creative text in text?

Creative text in sentence usually contains conceptual blending. Based on the LR finding, the selectional preference method proven as the most accurate and identifies wide range of creative text types. Selection preference method gives an accuracy of 80.84 %.

- b. Can system generate title with creative words and phrases; in a manner What is the best approach to generate creative title?

Framework generate title from creative sentences. It gets the sentences and title and creative sentences then it identifies the suitable creative sentence (template) for title. Finally, it substitutes words from suitable creative sentence and generate creative title. Lexical substitution was proven the best approach to generate creative title.

- c. Find suitable creative sentence for creative title

Finding suitable creative sentences is one of subtasks of creative title generation.

Similarity matrix is used to compare creative sentence and title based on the results top similar sentences will be chosen as candidates.

## 8.8 Critical Evaluation

Critical Evaluation concentrates on evaluating strengths and weaknesses of the model from the author's perspective

### 8.8.1 Evaluation on concept

Creative ideas and concepts are considered as important factor in many fields. The demand of creativity in fields such digital marketing, news and advertisement are high. Most online business uses creativity in different way to attract users to get their way. Some overusing this factors (Clickbait) and lose trust of their users and damage the brand name. Since high need of creativity areas, researches and projects in NLG and NLU grown past years.

To achieve creativity in the field tools such as slogan generator, title generators, catchy keyword analyzer and SEO analyzers are used. Automatic title generation important and complicated area in NLG. Creating title with creative word and phrase in it is an important study. Existing researches are mostly related to generate slogans, expression and idioms. Creative title generator would help content creators, advertisers, media and marketing areas. Issues current tools such abstract title generators are not matured to generate proper title from contents and also final outcome is not predictable which makes the system less trustable. Other than abstractive title generators slogans and expression generators are limited to its own templates (sentences be replaced) sources. It cannot identify suitable creative sentences to be used this will limit the scaling of the framework.

Creative text identification part helps to identify creative sentences in text. In fields like news outlets, advertising. They have previously used contents using this contents framework can identify creative phrase and sentences in it and reuse to generate title. This mean the framework could act as author when generating title by using used creative sentences before. This is an important feature of this framework. Also, it eliminates data preparation process of the framework. Existing frameworks source texts preparation took a long process by identifying suitable creative sentences type, required length and domains. Using creative sentences identification, a lot of time will by just passing set sentences it automatically derives creative sentences. Identification part not only identifies creative sentences for attractive users since, it identifies conceptual based sentences identified creative sentences would explain a concept in different domain.

With these features of framework an end application can be made with own customization and fulfil need in industry.

### 8.8.2 Evaluation on technical aspects

Framework build on light weight language Python. Python language supports for many NLP related libraries. Core functions are based on library spaCy. Spacy provide multi-processing features this work increases the performance of the framework. Also, it is identified as most accurate parser it increases the accuracy of framework. Selection of spacy it welcomed by

evaluators. Choosing sci-kit learn library for k-mean clustering is good choice in limited time clustering task completed with help of multi-processing feature. Choosing Glove embedding than Word2vec also good move because it worked well with selectional preference method. Gensim library used to load Word2vec embeddings it reduces the time of loading vectors.

Modularizing components made huge impact on decoupling components. Also, OOP based implementation helped reusability of objects. The module NLP used by identifier and generator module this shows the important of modules and OOP styles. The identification task implemented as pipeline style increases performance. spaCy uses multiprocessing with its pipeline architecture adding identification task as pipeline increases the performance speed. All the technical evaluators appreciated the selection of architectural styles and system design.

Standard coding practices followed to for better understanding. Python style guidance PEP 8 standard followed with documentation comments. Evaluators seems to be happy with it.

### **8.8.3 Evaluation of the Usefulness and Impact of framework**

Generating creative title with the framework is easy. The developer just has to setup creative text identification and creative title generation classes, the framework automatically identifies creative sentences and generate title using it. Thus, the developer will be more passionate about the framework. Generation and identification task go step by step so the developer can extend and make their own changes.

Furthermore, evaluators pointed out that title generation feature will be more useful in industries such as marketing and news. Evaluators also pointed out that it would be great if the title generators included SEO feature in keywords selection due to time constraint this feature is not developed in the initial prototype. But certainly, it will be considered in the future releases. In implemented stage some decisions affected the features. In k-mean clustering task, to achieve the wide coverage word groups recommended clusters amount is 300,000 with 400 iterations. Due to limited time clusters amount reduced to 200,000 with 300 iteration. But still it provided expected accuracy. Other one is, in identifying suitable candidates for title generation task the selection similarity matrix checking didn't give result as expected. After adding keyword word-based similarity implementation, it gave better results

Moreover, the accuracy level of creative text identification is fairly good. The accuracy of creative text identification is 80%. Also, performance of the system is also fine it identifies creative sentences within 2-3 minutes in 100 sentences. Accuracy and performance results of other generators are not published. Therefore, the performance results of the framework cannot be compared with others similar framework.

Test outcomes, quantitative assessment from the end-user's questionnaire, qualitative assessment from experts, and benchmarking proves positive outcomes. Therefore, the project can be decided as successful. Moreover, there are some improvements suggested by the evaluators which can be implemented in the future versions to make framework feature rich. Moreover, Future versions that can be combined into the initial prototype to increase the accuracy and performance.

## **8.9 Chapter Summary**

The project was evaluated in both quantitative and qualitative manners based on the evaluation goals. The marketing experts and software engineering domain experts evaluated the overall project. The general feedback of all the evaluators was positive. They also pointed out some areas that can be improved. Thereafter, the benchmarking was conducted to compare framework with previous implementations. Benchmarking revealed that framework stands in a good position compared to previous implementations. Then, the critical evaluation is conducted by the author and strengths and weaknesses of the project were discussed.

## **Chapter 9: Conclusion**



## 9. Conclusion

### 9.1 Chapter Overview

This chapter concludes the research by describing the achievement of the project. The chapter discusses achievement of the aim and objectives. It further explains the milestones and deliverables and how they were met in a timely manner. The problems and challenges faced during this project are also highlighted, and the solutions taken to overcome the issues are also briefly discussed. Furthermore, the future improvements that can be done to enhance the project are elaborated. The chapter concludes with concluding remarks.

### 9.2 Achievements of Objectives

Table 9.1 describes the achievement of the project objectives.

<b>Objective 1</b>	Study about creating creative titles.
<b>Achievement</b>	100%
<b>Remarks</b>	Studied about creating creative titles by examining popular articles, blog and news titles and identifies what kind of techniques and method used by them.
<b>Objective 2</b>	Research about title with different creative text.
<b>Achievement</b>	100%
<b>Remarks</b>	Studied about different title types and their impact on audience attraction. A thorough literature study was done, and the findings were documented in literature review.
<b>Objective 3</b>	Research about best way identify different types of creative texts.
<b>Achievement</b>	100%
<b>Remarks</b>	Researched about different types creative text identification method found most creative text contain conceptual blending. Based on that best approach to identify conceptual blending review on literature review.
<b>Objective 4</b>	Test identifying creative text identifier
<b>Achievement</b>	90%

<b>Remarks</b>	Due to limited amount of available dataset creative text identification accuracy tested on metaphors, expressions and clichés dataset. Performance of identification also tested. Test results and analysis documented in testing chapter.
<b>Objective 5</b>	Examine suitable data sources to be used
<b>Achievement</b>	100%
<b>Remarks</b>	Suitable data source for clustering and selectional preference identified by comparing available data sources. Finding were documented in literature review.
<b>Objective 6</b>	Research title generation from contents.
<b>Achievement</b>	100%
<b>Remarks</b>	Researched about different title generation approaches identified most suitable one for the project. Finding were documented in literature review.
<b>Objective 7</b>	Study about identifying keywords in title.
<b>Achievement</b>	100%
<b>Remarks</b>	Studied types of required keywords and identifying methods from exiting researches and examining titles from popular contents.
<b>Objective 8</b>	Research about adding and changing selected creative text in title without changing title's context.
<b>Achievement</b>	100%
<b>Remarks</b>	Based on findings lexical substitution was identified as suitable method. Finding were documented in literature review

*Table 9.1 - Achievements of objectives*

### 9.3 Milestones and Deliverables

#	Task Name	Estimated End	Actual End	Done (%)	Remarks
1	PID Document	04-11-18	04-11-18	100	The proposal was submitted on time

2	Literature review	03-01-19	18-03-19	100	Research papers and documents were referred till the implementation time to improve the domain knowledge.
3	Requirement Specification	21-01-19	01-02-19	80	Interviews were delayed so required more time to finish requirement specification
4	Design	05-02-19	12-02-19	88	Delay in requirement gathering
5	Implementation	13-03-19	05-03-19	95	Development was delayed due to the technical issues. luxury features are not implemented.
6	Testing & evaluation	28-03-19	28-03-19	95	Accuracy testing data set preparation took more time than planned
5	Project Closure	17-04-19	03-05-19	100	Submitted on time

Table 9.2 - Milestones deliverables

## 9.4 Achievement of Requirements

### 9.4.1 Achievements of functional requirements

#	Requirement	Done (%)	Remarks
FR1	Read corpus and extract contents	100	Completed
FR2	Process documents	100	Completed
FR3	Identify creative sentences	100	Completed
FR4	Generate title.	100	Completed
FR5	Find suitable templates for creative title.	100	Completed
FR6	Find important words in title and templates	100	Completed
FR7	Identify alias and pseudonym names	100	Completed

FR8	Generate creative title	100	Completed
FR9	Map alias/pseudonym with creative title	100	Completed
FR10	Validate generated creative title	0	Could not implement the feature due to limited time. Moreover, it was not an essential feature since it was a least prioritized functional requirement.
FR11	Check SEO analysis	0	Could not implement the feature due to limited time. Moreover, it was not an essential feature since it was a least prioritized functional requirement.

Table 9.3 - Achievements of functional requirements

#### 9.4.2 Achievements of non-functional requirements

ID	Requirement	Done (%)	Remarks
NFR1	Accuracy	86	The accuracy of the identification part tested and affirmed at 86%, which is considered as a good accuracy for the initial stage of the prototype. The accuracy of generation wasn't tested because measuring accuracy of creativity is difficult.
NFR2	Performance	70	The performance of identification and generation tested. overall performance is at an acceptable range for this project.
NFR3	Extendibility	80	The extendibility of the project is very high. The framework features are implemented considering extendibility.

Table 9.4 - Achievements of non-functional requirements

### 9.5 Achievement of Aim

***“The aim of this research project is to design & develop a title generating framework which attract users”.***

Initially, the problem domain and concept of the project were researched and identified via surveys, Interviews with domain experts and reviewing the literature. Therefore, it was favorable to gain the knowledge and idea of the identified problem. The core modules of the proposed solution were implemented using Python language. Then, the test outcomes and

evaluator's feedback illustrate that the framework succeeds appropriate approaches towards the problem of identifying and generating creative title. Therefore, the project achieved the aim successfully.

## 9.6 Limitations of the Project

- **Supported only for English Languages:** The initial prototype is only supported for the English language, but I can be extended simply with few tasks.
- **Some creative text won't be identified:** The framework's identification is based on conceptual blending creative types. There are some few types which doesn't have conceptual blending which won't be identified. But it doesn't affect identification task because there are not much used to attract users.
- **Generating title won't be entirely different than templates:** The generated title is something similar to creative sentences given to framework.
- **Identification part depending on library:** When extending and make customization on creative text identification developer has to depend on spaCy library because the identification task built upon it.

## 9.7 Learning Outcomes

- A prior knowledge of tools and technologies used in marketing, advertising area, Knowledge of NLP techniques and ML algorithms was gained. Moreover, learned technologies and tools utilized to develop the framework including Python, spaCy, genism and sci-kit learn.
- Architectural styles, design patterns, good coding practices and testing Frameworks were learnt and practiced.
- Documentation skill and academic writing skills were fairly developed.
- Critical Thinking, problem solving skills and time management skills were developed throughout the research.
- Implementing an existing algorithm from scratch.

## 9.8 Future Enhancements

<b>ID</b>	EH1	<b>Priority</b>	High
<b>Enhancement</b>	Generate reports		
<b>Description</b>	Generate identified creative text information from given sentences details.		
<b>ID</b>	EH2	<b>Priority</b>	Medium

<b>Enhancement</b>	Support different languages		
<b>Description</b>	Generate and identify creative sentences for different languages. By extending selectional preference data frequency and clustering to support other languages.		
<b>ID</b>	EH3	<b>Priority</b>	Medium
<b>Enhancement</b>	Reduce the memory taken by the framework		
<b>Description</b>	Reduce RAM space by load models on required time and unload other times.		

*Table 9.5 - Future enhancements*

## 9.9 Problems and Challenges Faced

- A Vast Research Area

NLU and NLG both has vast area to research. Since the project related with domains lexical substitution, title generation and selectional preference more than 30 research papers were studied to understand the project.

- Limited Time

The entire research was limited to 8 months of time. Accomplishing good results in such a short period of time in multiple research domains is highly challenging.

- Lack of experts

Lack of experts to talk on the research area or online help was another issue faced when implementing the framework since these were completely new technologies for the author.

## 9.10 Concluding Remarks

Due to lack of creativity tools to overcome competitors and attract target audience in fields such as advertisements and digital marketing the project propose a solution to gives creativity-based framework for help developers to setup end application. The framework provides features for identifying creative text, generate title and generate creative title. This would automate creative title generation work and improve the creativity.

## References

- Alfonseca, E., Pighin, D. and Garrido, G. (2013) 'HEADY: News headline abstraction through event pattern clustering', *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1243–1253.
- Ayana *et al.* (2017) 'Recent Advances on Neural Headline Generation', *Journal of Computer Science and Technology*, 32(4), pp. 768–784. doi: 10.1007/s11390-017-1758-3.
- Beliga, S. (2014) 'Keyword extraction: a review of methods and approaches', *University of Rijeka, Department of Informatics, Rijeka*, pp. 1–9.
- Blom, J. N. and Hansen, K. R. (2015) 'Click bait: Forward-reference as lure in online news headlines', *Journal of Pragmatics*, pp. 87–100. doi: 10.1016/j.pragma.2014.11.010.
- Brin, S. and Page, L. (1998) 'ScienceDirect - Computer Networks and ISDN Systems : The anatomy of a large-scale hypertextual Web search engine\*1', *Computer Networks and ISDN Systems*. doi: 10.1016/S0169-7552(98)00110-X.
- Brown, P. F. *et al.* (1992) 'Class-Based n-gram Models of Natural Language', *Association for Computational Linguistics*.
- Colmenares, C. A. *et al.* (2015) 'HEADS: Headline Generation as Sequence Prediction Using an Abstract Feature-Rich Space', *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 133–142. doi: 10.3115/v1/N15-1014.
- Van de Cruys, T. (2015) 'A Neural Network Approach to Selectional Preference Acquisition', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 26–35. doi: 10.3115/v1/d14-1004.
- Dhanya, P. M. (2013) 'Comparative Study of Text Summarization in Indian Languages', 75(6), pp. 17–21.
- Explosion AI (2019) *Language Processing Pipelines*. Available at: <https://spacy.io/usage/processing-pipelines> (Accessed: 1 May 2019).
- Fauconnier, G. and Turner, M. (2001) 'Conceptual Blending', *International Encyclopedia of the Social & Behavioral Sciences*, 19(2003), pp. 2495–2498. doi: 10.1016/B0-08-043076-7/00363-6.
- Fauconnier, G. and Turner, M. (2008) *The way we think: Conceptual blending and the mind's*

*hidden complexities. Basic Books.*

Fellbaum, C. (2010) 'WordNet', in *Theory and Applications of Ontology: Computer Applications*. doi: 10.1007/978-90-481-8847-5\_10.

Frampton, B. (2015) 'Clickbait: The changing face of online journalism', *BBC*, (September), pp. 1–3. Available at: <https://www.bbc.com/news/uk-wales-34213693> (Accessed: 13 December 2018).

Gambhir, M. and Gupta, V. (2017) 'Recent automatic text summarization techniques: a survey', *Artificial Intelligence Review*. Springer Netherlands, 47(1). doi: 10.1007/s10462-016-9475-9.

Gattani, A. K. (2007) *Automated Natural Language Headline Generation Using Discriminative Machine Learning Models*.

Gatti, L. *et al.* (2015) 'Slogans are not forever: Adapting linguistic expressions to the news', in *IJCAI International Joint Conference on Artificial Intelligence*, pp. 2452–2458.

Haagsma, H. and Bjerva, J. (2016) 'Detecting novel metaphor using selectional preference information', *Proceedings of The Fourth Workshop on Metaphor in NLP*, (June), pp. 10–17. doi: 10.18653/v1/W16-1102.

Hawtin, A. (2018) 'The British National Corpus Revisited: Developing parameters for Written BNC2014'. Available at: <http://www.birmingham.ac.uk/Documents/college-artslaw/corpus/conference-archives/2017/general/paper39.pdf>.

Jin, R. (2003) *Statistical Approaches Toward Title Generation*. Carnegie Mellon University. doi: 10.16309/j.cnki.issn.1007-1776.2003.03.004.

Kinzler, D. and Pintscher, L. (2014) 'Wikidata', *Proceedings of The International Symposium on Open Collaboration - OpenSym '14*, pp. 1–1. doi: 10.1145/2641580.2641583.

Knight, K. and Marcu, D. (2000) 'Statistics-Based Summarization - Step One: Sentence Compression', *Aaai*, pp. 703–710. doi: [http://dx.doi.org/10.1016/S0004-3702\(02\)00222-9](http://dx.doi.org/10.1016/S0004-3702(02)00222-9).

Kremer, G. *et al.* (2015) 'What Substitutes Tell Us - Analysis of an "All-Words" Lexical Substitution Corpus', pp. 540–549. doi: 10.3115/v1/e14-1057.

Light, M. and Greiff, W. (2002) 'Statistical models for the induction and use of selectional preferences', *Cognitive Science*, 26(3), pp. 269–281. doi: 10.1016/S0364-0213(02)00070-8.

Lucy, B. (2007) *End of the dream for migrant builder who lost leg in cave in* | *The Times*. Available at: <https://www.thetimes.co.uk/article/end-of-the-dream-for-migrant-builder-who>



lost-leg-in-cave-in-gp7cj72mxcn (Accessed: 3 November 2018).

McCarthy, D. and Navigli, R. (2007) 'SemEval-2007 Task 10: English Lexical Substitution Task', *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 48–53. doi: 10.1007/s10579-009-9084-1.

McQuarrie, E. F. and Mick, D. G. (1992) 'On Resonance: A Critical Pluralistic Inquiry Into Advertising Rhetoric', *Journal of Consumer Research*. doi: 10.1086/209295.

Mehdi, Allahyari *et al.* (2018) 'Text summarization techniques: a brief survey.', in. doi: 10.1145/nnnnnnn.nnnnnnn.

Melamud, O., Levy, O. and Dagan, I. (2015) 'A Simple Word Embedding Model for Lexical Substitution', in, pp. 1–7. doi: 10.3115/v1/w15-1501.

Mihalcea, R. and Tarau, P. (2004) 'TextRank: Bringing Order into Texts', *Proceedings of the 2004 conference on empirical methods in natural language processing*, (4). doi: 10.1016/0305-0491(73)90144-2.

Mihalcea, R. and Tarau, P. (2005) 'A Language Independent Algorithm for Single and Multiple Document Summarization', *Proceedings of IJCNLP 2005, 2nd International Joint Conference on Natural Language Processing*, pp. 19–24.

Mikolov, T. *et al.* (2013) 'Efficient Estimation of Word Representations in Vector Space', pp. 1–12. Available at: <http://arxiv.org/abs/1301.3781>.

Munigala, V. *et al.* (2018) 'PersuAIDE! An Adaptive Persuasive Text Generation System for Fashion Domain Vitobha', *In Companion of the The Web Conference 2018 on The Web Conference 2018. International World Wide Web Conferences Steering Committee*, 2226, pp. 335–342. doi: 10.1145/nnnnnnn.nnnnnnn.

Muzny, G. and Zettlemoyer, L. (2012) 'Automatic Idiom Identification in Wiktionary'.

Nathan, S. (2013) *5 Data Insights into the Headlines Readers Click - Moz*. Available at: <https://moz.com/blog/5-data-insights-into-the-headlines-readers-click> (Accessed: 29 October 2018).

Özbal, G., Pighin, D. and Strapparava, C. (2013) 'BrainSup: Brainstorming support for creative sentence generation', *ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 1, pp. 1446–1455. doi: 10.1097/MCP.0000000000000056.

Pan, B. *et al.* (2007) 'In Google we trust: Users' decisions on rank, position, and relevance', *Journal of Computer-Mediated Communication*, 12(3), pp. 801–823. doi: 10.1111/j.1083-

6101.2007.00351.x.

Pennington, J., Socher, R. and Manning, C. (2014) 'Glove: Global Vectors for Word Representation', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. doi: 10.3115/v1/D14-1162.

Phillips, B. J. and McQuarrie, E. F. (2009) 'Impact of Advertising Metaphor on Consumer Belief: Delineating the Contribution of Comparison Versus Deviation Factors', *Journal of Advertising*, 38(1), pp. 49–62. doi: 10.2753/JOA0091-3367380104.

Piotrkowicz, A. (2017) *Modelling Social Media Popularity of News Articles Using Headline Text*.

Radev, D. R. and Erkan, G. (2004) 'LexRank: Graph-based Centrality as Saliency in Text Summarization', *Journal of Artificial Intelligence Research*, 22(1), pp. 457–479. doi: 10.1613/jair.1523.

Resnik, P. (1993) 'Selection and information: a class-based approach to lexical relationships', *IRCS Technical Reports Series*, pp. 93–42. Available at: [http://repository.upenn.edu/cgi/viewcontent.cgi?article=1192&context=ircs\\_reports](http://repository.upenn.edu/cgi/viewcontent.cgi?article=1192&context=ircs_reports).

Resnik, P. (1996) 'Selectional constraints: an information-theoretic model and its computational realization', *Cognition*. doi: 10.1016/S0010-0277(96)00722-6.

Rooth, M. *et al.* (1999) 'Inducing a Semantically Annotated Lexicon via EM-Based Clustering', pp. 104–111. doi: 10.3115/1034678.1034703.

Said-Metwaly, S., Noortgate, W. Van den and Kyndt, E. (2018) 'Approaches to Measuring Creativity: A Systematic Literature Review', *Creativity. Theories – Research - Applications*, 4(2), pp. 238–275. doi: 10.1515/ctra-2017-0013.

Scikit-Learn Mini-Batch K-Means (2019) *Comparison of the K-Means and MiniBatchKMeans clustering algorithms — scikit-learn 0.19.1 documentation*. Available at: [https://scikit-learn.org/stable/auto\\_examples/cluster/plot\\_mini\\_batch\\_kmeans.html](https://scikit-learn.org/stable/auto_examples/cluster/plot_mini_batch_kmeans.html) (Accessed: 2 May 2019).

Shao, L. and Wang, J. (2017) 'DTATG: An Automatic Title Generator based on Dependency Trees'. Available at: <http://arxiv.org/abs/1710.00286>.

Shu, K. *et al.* (2015) 'Deep Headline Generation for Clickbait Detection'.

Shutova, E., Sun, L. and Korhonen, A. (2010) 'Metaphor Identification Using Verb and Noun Clustering', *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, (August), pp. 1002–1010. Available at:

<http://www.cl.cam.ac.uk/~es407/papers/Coling10.pdf>.

Shutova, E., Teufel, S. and Korhonen, A. (2013) 'Statistical Metaphor Processing', (April 2012). doi: 10.1162/COLI.

Simon Kemp (2018) *Digital in 2018: World's internet users pass the 4 billion mark - We Are Social, We Are Social*. Available at: <https://wearesocial.com/blog/2018/01/global-digital-report-2018> (Accessed: 29 October 2018).

Slant (2019) *39 Best Python IDEs or editors as of 2019 - Slant*. Available at: <https://www.slant.co/topics/366/~best-python-ides-or-editors> (Accessed: 22 April 2019).

Steen, G. J. *et al.* (2010) *A Method for Linguistic Metaphor Identification*. doi: 10.1075/celcr.14.

Szymanski, T., Orellana-Rodriguez, C. and Keane, M. T. (2017) 'Helping News Editors Write Better Headlines: A Recommender to Improve the Keyword Contents & Shareability of News Headlines'. Available at: <http://arxiv.org/abs/1705.09656>.

Toivanen, J. M. *et al.* (2012) 'Corpus-Based Generation of Content and Form in Poetry', *Proceedings of the Third International Conference on Computational Creativity*, pp. 175–179.

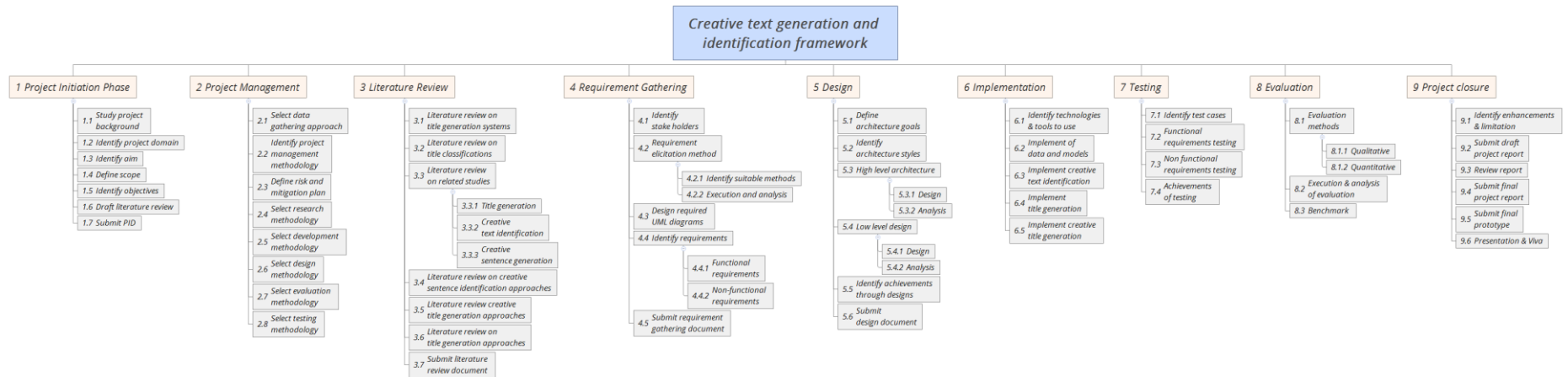
Unno, Y. *et al.* (2006) 'Trimming CFG parse trees for sentence compression using machine learning approaches', *Proceedings of the COLING/ACL on Main conference poster sessions* -, (September 2014), pp. 850–857. doi: 10.3115/1273073.1273182.

Valitutti, A., Stock, O. and Strapparava, C. (2009) 'GraphLaugh: A tool for the interactive generation of humorous puns', *Proceedings - 2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops, ACII 2009*, pp. 9–10. doi: 10.1109/ACII.2009.5349529.

Verma, R. and Vuppuluri, V. (2015) 'A New Approach for Idiom Identification Using Meanings and the Web', *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pp. 681–687. Available at: <http://aclweb.org/anthology/R15-1087>.

Wilks, Y. (1978) 'Making preferences more active', *Artificial Intelligence*. doi: 10.1016/0004-3702(78)90001-2.

# Appendix A Word Breakdown Structure



## Appendix B Interviewees

#	Qualification	Designation
1	MSc	Tech lead at Aeturnum Lanka
2	ACIM	Digital Marketer at Ekwa marketing

## Appendix C Use Case Descriptions

<b>Use Case ID</b>	UC4
<b>Use Case Name</b>	Process Document
<b>Priority</b>	High
<b>Participating Actors</b>	None
<b>Precondition</b>	Sentences or text should be provided
<b>Postcondition</b>	
<b>Included Use Case</b>	None
<b>Triggering event</b>	
<b>Description</b>	Does text processing and NLP related work
<b>Main Flow</b>	
<ol style="list-style-type: none"> <li>1. Validate sentence</li> <li>2. Segment document</li> <li>3. Remove digits and symbols in sentences</li> <li>4. Remove stop words from sentences</li> <li>5. Strip white spaces in sentences</li> <li>6. Tokenize document</li> <li>7. Assign part-of-speech tags for document.</li> <li>8. Vectorize words</li> <li>9. Lemmatize words</li> <li>10. Find named entities</li> <li>11. Dependency parse document</li> </ol>	
<b>Alternative Flow</b>	

- Alternative Flow 1  
At step 8. If vector for not found for word  
Find vector for closest similar word

### **Exceptional Flow**

- Expectational Flow 1  
At step 8. If vector for not found for word and similar word  
Mark as empty

<b>Use Case ID</b>	UC5
<b>Use Case Name</b>	Read corpus
<b>Priority</b>	High
<b>Participating Actors</b>	None
<b>Precondition</b>	Required a corpus path
<b>Postcondition</b>	None
<b>Included Use Case</b>	None
<b>Triggering event</b>	Called by identifier or generator
<b>Description</b>	Read corpus and return sentences
<b>Main Flow</b>	
<ol style="list-style-type: none"> <li>1. Open directory path</li> <li>2. Collection files in the directory</li> <li>3. Read files</li> <li>4. Segment files</li> </ol>	
<b>Exceptional Flow</b>	
<ul style="list-style-type: none"> <li>• Exceptional Flow 1 At step 2, If directory not found Display not found error message</li> <li>• Exceptional Flow 2 At step 3, If files not found Display not found error message</li> </ul>	

<b>Use Case ID</b>	UC6
<b>Use Case Name</b>	Generate Title
<b>Priority</b>	High
<b>Participating Actors</b>	
<b>Precondition</b>	Required ratio or length for title should be provided
<b>Postcondition</b>	None
<b>Included Use Case</b>	None
<b>Triggering event</b>	Application Developer execute generation task
<b>Description</b>	Find keywords for given sentence
<b>Main Flow</b>	
<ol style="list-style-type: none"> <li>1. Read text and tokenize</li> <li>2. Generate Similarly Matrix across sentences</li> <li>3. Rank sentences in similarity matrix</li> <li>4. Sort the rank and pick top sentences using TextRank</li> <li>5. Generate title based on given ration or length</li> </ol>	
<b>Alternative Flow</b>	
<ul style="list-style-type: none"> <li>• Alternative Flow 1 At step 3. If required ratio or length not given Display error message</li> </ul>	

<b>Use Case ID</b>	UC7
<b>Use Case Name</b>	Generate Creative Title
<b>Priority</b>	High
<b>Participating Actors</b>	Application Developer
<b>Precondition</b>	Titles and creative text templates should be provided
<b>Postcondition</b>	Provide creative title for document



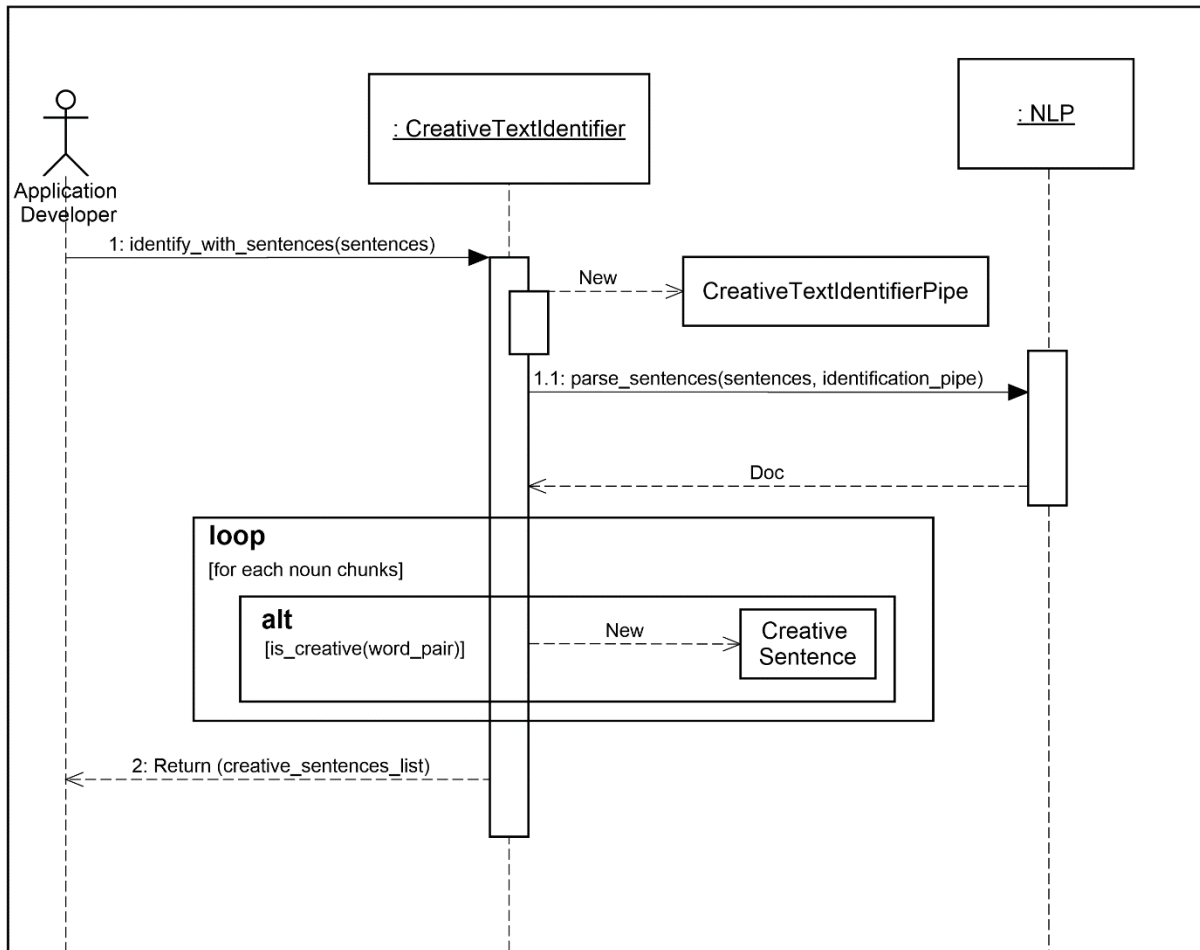
<b>Included Use Case</b>	Find suitable templates, Process corpus
<b>Triggering event</b>	Application Developer execute generation task
<b>Description</b>	It requires two things title and creative text templates. It identifies suitable template to for title and replaces words to generate new creative title.
<b>Main Flow</b>	
<ol style="list-style-type: none"> <li>1. Setup title and creative sentences templates</li> <li>2. &lt;&lt;include&gt;&gt; Process document</li> <li>3. &lt;&lt;include&gt;&gt; Find keywords</li> <li>4. Find similar and substitutable words for keywords</li> <li>5. &lt;&lt;include&gt;&gt; Find suitable templates</li> <li>6. Sort and pick candidate templates</li> <li>7. Calculate score for suitable replace words in template with title keywords</li> <li>8. Generate creative title</li> </ol>	
<b>Exceptional Flow</b>	
<ul style="list-style-type: none"> <li>• Exceptional Flow 1 At step 4, If similar and substitute words not found Display not found error message</li> <li>• Exceptional Flow 2 At step 5, If creative text templates not found Display error message</li> <li>• Exceptional Flow 3 At step 7, if score not satisfied Display suitable word not found error message</li> </ul>	

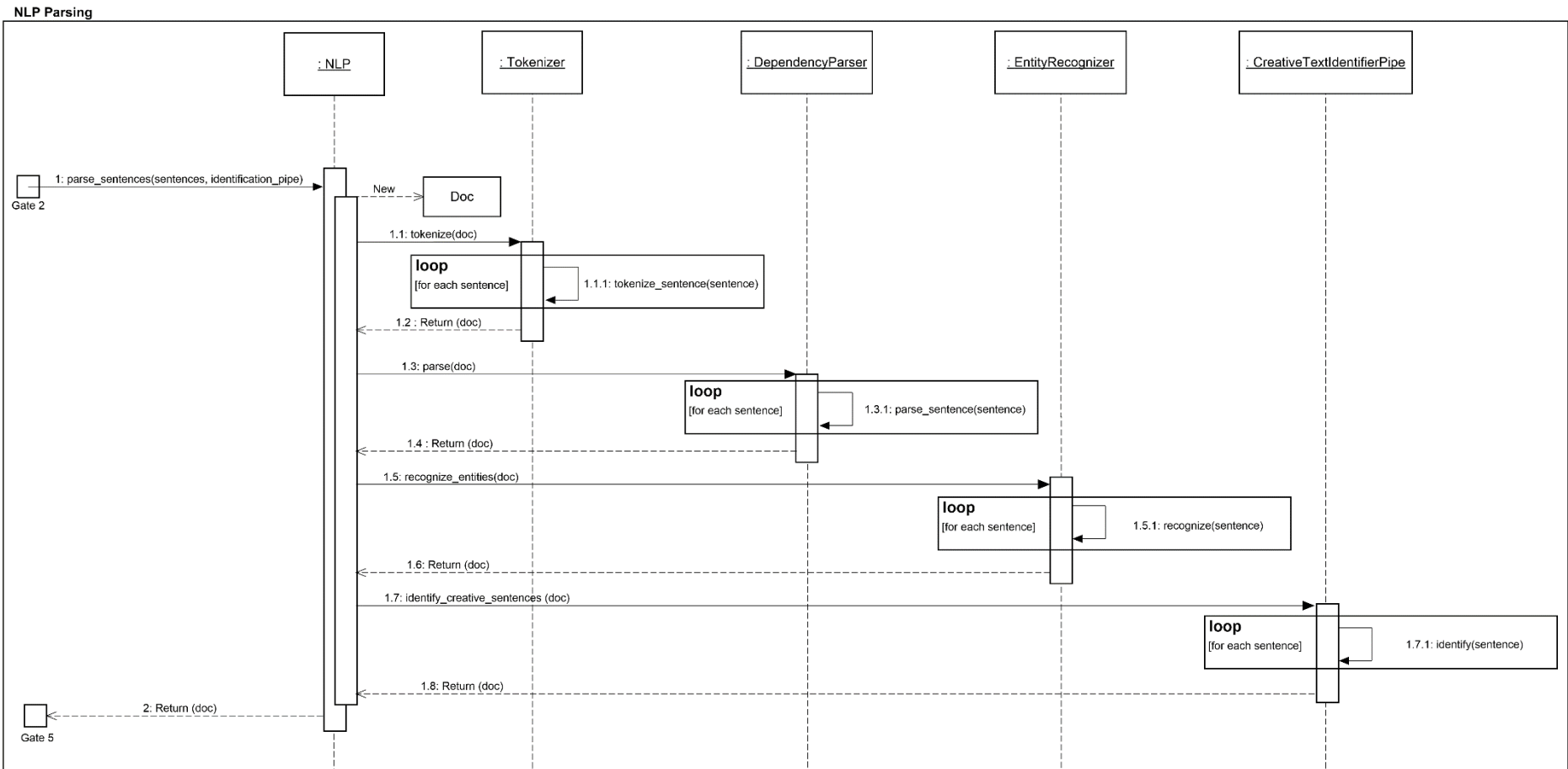
<b>Use Case ID</b>	UC8
<b>Use Case Name</b>	Find keywords
<b>Priority</b>	High
<b>Participating Actors</b>	None
<b>Precondition</b>	<ul style="list-style-type: none"> <li>Given sentence should be processed in UC3 use case</li> <li>Word frequency model should be loaded</li> </ul>
<b>Postcondition</b>	None
<b>Included Use Case</b>	None
<b>Triggering event</b>	UC7 call the task
<b>Description</b>	Find keywords in given document
<b>Main Flow</b>	
<ol style="list-style-type: none"> <li>1. Identify valid words in sentence</li> <li>2. Get frequency of words</li> <li>3. Calculate word scoring</li> <li>4. Check score greater than important word threshold value</li> </ol>	
<b>Exceptional Flow</b>	
<ul style="list-style-type: none"> <li>Exceptional Flow 1 At step 1, If validate words not found Display not found error message</li> <li>Exceptional Flow 2 At step 1, If frequency of word not found Set as empty and continue</li> </ul>	

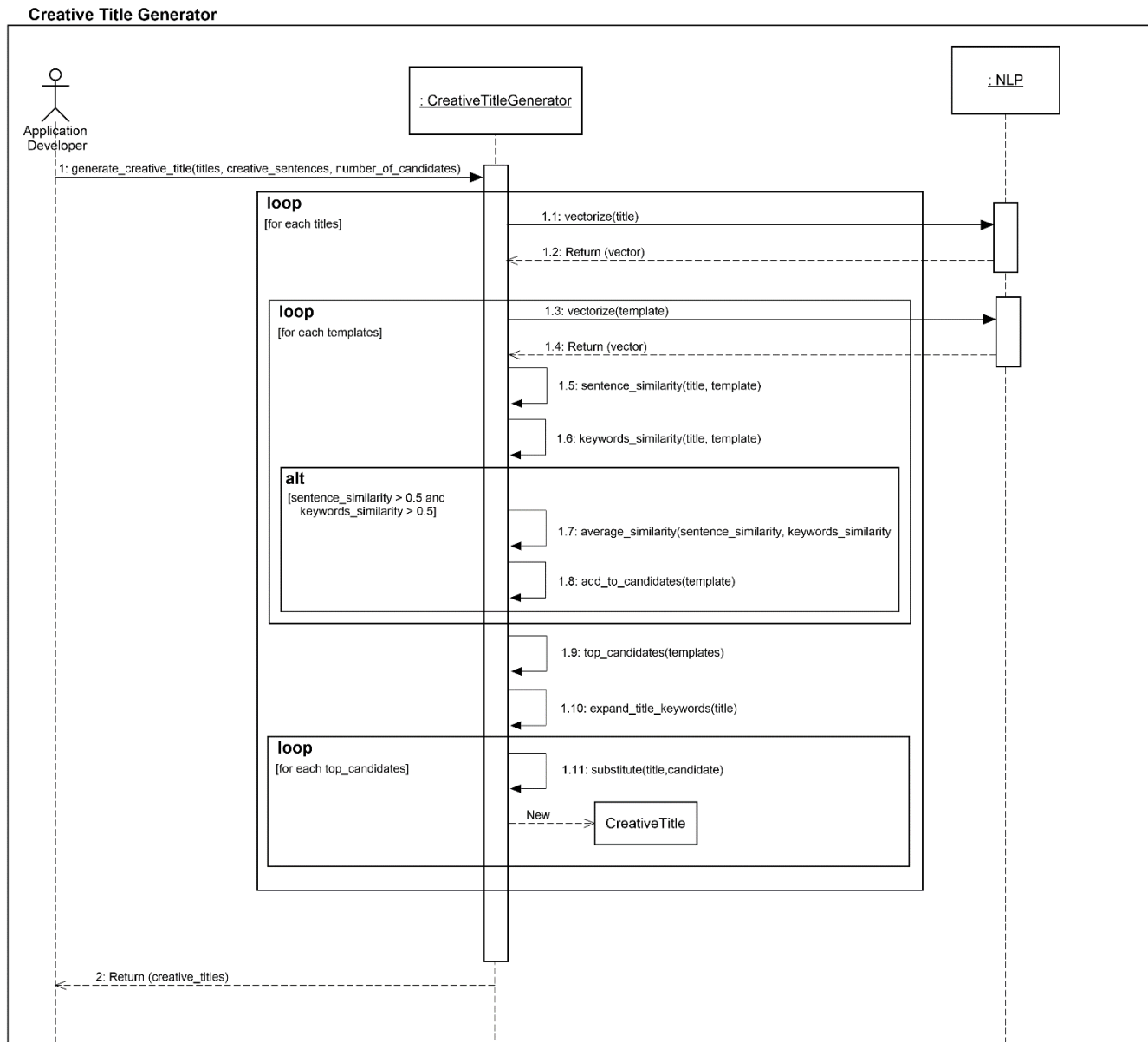
<b>Use Case ID</b>	UC9
<b>Use Case Name</b>	Find suitable templates
<b>Priority</b>	High
<b>Participating Actors</b>	None
<b>Precondition</b>	Creative text templates should be provided
<b>Postcondition</b>	None
<b>Included Use Case</b>	Find Keywords
<b>Triggering event</b>	UC7 call the task
<b>Description</b>	Find suitable creative text templates for creative title generation use case
<b>Main Flow</b>	
<ol style="list-style-type: none"> <li>1. &lt;&lt;include&gt;&gt; Find Keywords for title and templates</li> <li>2. Find similar words for keywords of title and templates</li> <li>3. Vectorize keywords and similar words of title and templates</li> <li>4. Calculate cosine similarity between title and template sentence vectors</li> <li>5. Calculate cosine similarity between title and template keywords vectors</li> <li>6. Check score are greater threshold value</li> </ol>	
<b>Exceptional Flow</b>	
<ul style="list-style-type: none"> <li>• Exceptional Flow 1 <ul style="list-style-type: none"> <li>At step 2, If similar words not found for keywords <ul style="list-style-type: none"> <li>Display not found error message</li> </ul> </li> </ul> </li> </ul>	

## Appendix D Sequence Diagrams

### Creative Text Identification







## Appendix E Evaluators

#	Qualification	Designation
1	BEng	Tech lead at Aeturnum Lanka
2	MSc	Tech lead at Aeturnum Lanka
3	BSc	Associate tech lead at Aeturnum Lanka
4	ACIM	Digital Marketer at Ekwa marketing
5	BA (Hons)	Head of Marketing at Ekwa

## Appendix F Unit Test cases

#	Test Case Scenario	FR ID	Priority	Input Type	Actual Output	Expected Output	Result	Remark
1	Framework receive corpus read object	FR1	Low	Root folder path and encoding type	Corpus Read object	Corpus Read object	Pass	Tested 2 times
2	Framework receive corpus read object with invalid path	FR1	Low	Root folder path and encoding type	File not found exception	File not found exception	Pass	Usual
3	Framework receive input as sentences and passed process sentences	FR2	High	Set of sentences	Doc object with parsed sentences	Doc object with parsed sentences	Pass	Usual
4	Framework receive input as corpus reader object and passed process sentences	FR2	High	Corpus Reader object	Doc object with parsed sentences	Doc object with parsed sentences	Pass	Usual
5	Framework receive input as corpus reader object with no sentences and passed to process sentences	FR2	High	Corpus Reader Object	No sentences exception	No sentences exception	Pass	Usual



6	Framework receive input as Tamil sentences and passed process sentences	FR2	High	Set of sentences	Not supported language exception	Not supported language exception	Pass	Usual
7	Framework received doc object with creative sentences and pass to identify creative sentences method	FR3	High	Doc object	Creative Sentence Object	Creative Sentence Object	Pass	Tested 3 times with different inputs
8	Framework received doc object without creative sentences and pass to identify creative sentences method	FR3	High	Doc object	Empty array	Empty array	Pass	Tested 3 times with different inputs
9	Pass a word in Glove vector for vectorizing process	FR3	High	Words	Numpy Array	Numpy Array	Pass	Tested 10 times with different inputs
10	Pass a word for vectorizing process which doesn't exist in Glove vectors, but similar word does exist	FR3	High	Words	Numpy Array vector of most similar word	Numpy Array vector of most similar word	Pass	Tested 10 times with different inputs

11	Pass a word for vectorizing process which doesn't exist in Glove vectors and its similar word	FR3	High	Word	KeyError exception	KeyError exception	Pass	Tested 10 times with different inputs
12	Pass word vector for clustering	FR3	High	[ 7.2109e-03 - 2.5406e-01 -1.41... Vector numpy array	Cluster groups	Cluster groups	Pass	Tested 5 times with different inputs
13	Pass verb cluster which is available in word frequency for SPS calculation	FR3	High	Cluster groups	SPS float value	SPS float value	Pass	Tested 5 times with different inputs
14	Pass verb cluster which is not available in word frequency for SPS calculation	FR3	High	Cluster groups	Return None value	Return None Value	Pass	Tested 5 times with different inputs
15	Pass verb and noun clusters which are available in word frequency for SA calculation	FR3	High		SA float value	SA float value	Pass	Tested 5 times with different inputs

16	Finding word is creative with SA values	FR3	High		Return True value	Return True Value	Pass	Tested 5 times with different inputs
17	Framework received sentences for generate title	FR4	High	List of sentences	Title Object	Title Object	Pass	Tested 3 times with different inputs
18	Framework received one sentence for generate title	FR4	High	A List with one sentence	Value Error	Value Error	Pass	Tested 1 time
19	Framework received sentences in Tamil language for generate title	FR4	High	List of sentences	Not supported language exception	Title Object	Fail	Tested 1 time
20	Framework received creative sentences object and title object for creative title generation method	FR8	High	Creative sentences List and Title List	Creative Title Object	Creative Title Object	Pass	Usual
21	Framework received empty creative sentences object and	FR8	High	Empty Lists	Value Error Exception	Value Error Exception	Pass	Usual

	empty title object for creative title generation method							
<b>22</b>	Finding similar words in finding suitable candidate for title	FR8	High	Import token list	Word, Word list dictionary	Word, Word list dictionary	Pass	Usual
<b>23</b>	Framework find suitable creative sentence candidates for title in creative title process	FR5	High	Title object and Creative sentence List	Creative sentence List	Creative sentence List	Pass	Tested 5 times with different inputs
<b>24</b>	Framework find suitable creative sentence candidates without parsing for title in creative title process	FR5	High	Title object and Creative sentence List	Creative sentence List	Creative sentence List	Pass	Tested 2 times with different inputs
<b>25</b>	Sentence similarity check in finding suitable candidates for title	FR5	High	Title text and template text	Boolean value	Boolean value	Pass	Usual
<b>26</b>	Keyword similarity check in finding suitable candidates for title	FR5	High	Title important token list and template important token list	Boolean value	Boolean value	Pass	Usual

<b>27</b>	Framework find important words in title and templates in creative title process	FR6	High	Title object and Creative sentence object	Word index list	Word index list	Pass	Usual
<b>28</b>	Framework find alias and pseudonym names for word	FR7	Low	Token text	Text List	Text List	Pass	Usual
<b>29</b>	Framework insert or replace pseudonym name in title object	FR9	Low	Index, Word Dictionary	Boolean and index tuple	Boolean and index tuple	Pass	Usual

## Appendix G Module Integration Test

#	Module	Test Case	Input	Actual Output	Expected Output	Result
1	NLP Module	Read contents from corpus reader	Root directory path	File contents	File Contents	Pass
2	NLP Module	Identify substitute words in sentence	Token and sentence	List of words	List of words	Pass
3	NLP Module	Parse sentence	Sentence List	Doc object with parsed sentences	Doc object with parsed sentences	Pass
4	Identifier module	SA calculation on word frequency	Word frequency model and verb	SPS float value	SPS float value	Pass
5	Identifier module, NLP module	Creative text identification from raw sentence list	Sentence List	Creative Sentence List	Creative Sentence List	Pass
6	Identifier module, NLP module	Creative text identification from corpus reader object	Corpus Reader object	Creative Sentence List	Creative Sentence List	Pass
7	Generator module	Generate title from sentences	Sentence List	Title object	Title object	Pass
8	Generator module, NLP module	Generate title from corpus reader object	Corpus reader object	Title object	Title object	Pass

9	Generator module, NLP module	Creative title generation	List of title object and list of creative sentences	Creative title object	Creative title object	Pass
---	------------------------------	---------------------------	---	-----------------------	-----------------------	------

## Appendix H Title template word pair

#	Title	Templates
1	The Obama administration is planning to issue a final rule designed to enhance the safety of offshore oil drilling equipment.	Bridge over troubled water
2	Russia's defense ministry has rejected complaints by U.S. officials who claimed Russian attack planes buzzed dangerously close to a U.S. Navy destroyer[...]	The empire strikes back
3	There will be no soft Brexit now. It's no deal, revoke or another vote	Throw cold water on
4	Doctor describes 'ecstatic' moment coma patient woke up after 27 years	He's waiting for his ship to come in
5	Eden Hazard snubbed for Player of the Year	Turn a blind eye

## Appendix I Random Templates

#	Random Templates
1	Wash your hands of something
2	Made it by the skin of my teeth
3	Buy something for a song
4	We're going to burn the midnight oil
5	Catch hell if I do
6	It's a game of inches
7	we're not in Kansas anymore
8	lift your game (get your act together)
9	doing the horizontal bop



<b>10</b>	If you're going to talk the talk, you better walk the walk.
<b>11</b>	Throw cold water on
<b>12</b>	A house divided against itself cannot stand.
<b>13</b>	the butler did it
<b>14</b>	Bird in the hand is worth two in the bush.
<b>15</b>	putting the cart before the horse
<b>16</b>	all the world's a stage
<b>17</b>	Never cast a cloud till May be out
<b>18</b>	I have not slept one wink.
<b>19</b>	it's all in your head
<b>20</b>	Make a virtue of necessity

## Appendix J Finding keywords

#	Sentence	Keywords picked by individuals	Keywords picked by framework	Summed keywords	Accuracy (%)
1	santos says mud disaster funds appropriate	Santos, funds, appropriate	Santos, appropriate, disaster	1	33
2	firefighters join vic fire effort	Firefighters, join, effort	Firefighters, join, effort, fire	2	100
3	cleaners march through cbd over pay conditions	Cleaners, pay, conditions	Cleaners, pay	2	67
4	hurricane ivan kills 10 in Caribbean	Ivan, kills, Caribbean	Ivan, kills, Caribbean	3	100
5	tonnes of oil blanket queensland beaches	Oil, blanket, queensland	Oil, queensland	2	67
6	large two storey factory engulfed flames at williamstown north	Factory, flames, north	Large, factory, flames, williamstown	0	0
7	rain brings welcome relief for firefighters	Rain, welcome, firefighters	Rain, brings, firefighters	1	33
8	high winds ground balloon championships	Balloon, championships	High, balloon, championships	1	50

9	costello defends future fund move	Costello, defends, fund, move	Costello, defends, fund, move	4	100
10	children injured in train ride accident	Children, injured, accident	Children, injured, accident	3	100

## Appendix K Activity Schedule

Task Name	Start Date	End Date	Duration	Q4			Q1			Q2			Q3							
				Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun					
<b>FYP Project</b>	<b>08/08/18</b>	<b>06/05/19</b>	<b>306</b>																	
<b>Project Initiation Phase</b>	<b>08/08/18</b>	<b>04/11/18</b>	<b>77</b>																	
Identify The Problem Background	20/08/18	29/08/18	10																	
Identify Project Domain	30/08/18	09/09/18	11																	
Analyze Related Works	10/09/18	17/09/18	8																	
Identify Research Gap	18/09/18	25/09/18	8																	
Propose Solution for Identified Gap	26/09/18	03/10/18	8																	
Identify Aims & Scope	04/10/18	10/10/18	7																	
Identify Methodologies	11/10/18	15/10/18	5																	
Identify Objectives	16/10/18	21/10/18	6																	
Identify Requirements	22/10/18	28/10/18	7																	
Identify Data Gathering	29/10/18	31/10/18	3																	
Finalize Document	01/11/18	01/11/18	1																	
Submit Draft PID & Get Feedback	02/11/18	02/11/18	1																	
Motify PID Per Feedback	03/11/18	03/11/18	1																	
Submit Final PID	04/11/18	04/11/18	1																	
<b>Literature Review</b>	<b>05/11/18</b>	<b>03/01/19</b>	<b>60</b>																	
Literature Review On Automatic title generation systems	05/11/18	07/11/18	3																	
Literature Review On Related Studies	08/11/18	27/11/18	20																	
Literature Review On Title generation approaches	28/11/18	05/12/18	8																	
Literature Review On Creative text identification approaches	06/12/18	19/12/18	14																	
Literature Review On Creative text generation approaches	20/12/18	03/01/19	15																	
<b>Requirement Specification</b>	<b>04/01/19</b>	<b>21/01/19</b>	<b>19</b>																	
Requirement Gathering	06/01/19	09/01/19	4																	
Identify Stakeholders	10/01/19	13/01/19	4																	
Identify Requirements	14/01/19	16/01/19	3																	

	Task Name	Start Date	End Date	Duration	Q2		
					Jan	Feb	Mar
27	Draw UML Diagrams	17/01/19	23/01/19	7	█	Draw UML Diag	
28	Submit Requirement Specification	24/01/19	24/01/19	1	█	Submit Requirr	
29	<b>Design Specification</b>	<b>22/01/19</b>	<b>03/02/19</b>	<b>13</b>	█	<b>Design Sp</b>	
30	Define and Design architecture goals	22/01/19	22/01/19	1	█	Define and Des	
31	Identify Architecture styles	23/01/19	23/01/19	1	█	Identify Archite	
32	Design High Level Designs	24/01/19	28/01/19	5	█	Design High L	
33	Design Low Level Diagrams	29/01/19	02/02/19	5	█	Design Low l	
34	Submit design specification	03/02/19	03/02/19	1	█	Submit desig	
35	<b>Implementation</b>	<b>23/01/19</b>	<b>13/03/19</b>	<b>50</b>	█	<b>Im</b>	
36	<b>Title Generation</b>	<b>23/01/19</b>	<b>01/02/19</b>	<b>10</b>	█	<b>Title Genera</b>	
37	Implement title generation	23/01/19	27/01/19	5	█	Implement titl	
38	Prototype improvement	28/01/19	28/01/19	1	█	Prototype imp	
39	Identify test cases	29/01/19	29/01/19	1	█	Identify test c	
40	Unit test for title generation	30/01/19	30/01/19	1	█	Unit test for t	
41	Non-functional test	31/01/19	31/01/19	1	█	Non-function	
42	Test result analysis	01/02/19	01/02/19	1	█	Test result a	
43	<b>Creative Text Identifier</b>	<b>02/02/19</b>	<b>21/02/19</b>	<b>20</b>	█	<b>Creativ</b>	
44	Implement creative text identification	02/02/19	11/02/19	10	█	Implement	
45	Prototype improvement	12/02/19	14/02/19	3	█	Prototype	
46	Identify test cases	15/02/19	15/02/19	1	█	Identify t	
47	Unit test for title generation	16/02/19	17/02/19	2	█	Unit test	
48	Non-functional test	18/02/19	19/02/19	2	█	Non-fun	
49	Test result analysis	20/02/19	21/02/19	2	█	Test res	
50	<b>Creative Title Generator</b>	<b>22/02/19</b>	<b>13/03/19</b>	<b>20</b>	█	<b>Cre</b>	
51	Implement creative title generation	22/02/19	01/03/19	8	█	Implei	
52	Prototype improvement	02/03/19	06/03/19	5	█	Prot	
53	Identify test cases	07/03/19	07/03/19	1	█	Ider	
54	Unit test for title generation	08/03/19	09/03/19	2	█	Uni	

Task Name	Start Date	End Date	Duration	Q2			Q3		
				Jan	Feb	Mar	Apr	May	Jun
55 Non-functional test	10/03/19	11/03/19	2			█	█		
56 Test result analysis	12/03/19	13/03/19	2			█	█		
57 <b>Testing</b>	<b>14/03/19</b>	<b>15/03/19</b>	<b>1</b>			█	█		
58 <b>Evaluation</b>	<b>16/03/19</b>	<b>28/03/19</b>	<b>14</b>			█	█	█	█
59 Select evaluation method	15/03/19	17/03/19	3			█	█		
60 Apply evaluation method	18/03/19	27/03/19	10			█	█	█	█
61 Compare similar systems	28/03/19	28/03/19	1			█	█		
62 <b>Project Closure</b>	<b>29/03/19</b>	<b>17/04/19</b>	<b>20</b>			█	█	█	█
63 Identify Enhancements & Limitations	29/03/19	01/04/19	4			█	█		
64 Submit Draft Project Report & Get Feedback	02/04/19	02/04/19	1			█	█		
65 Modify Project Report per Feedback	03/04/19	15/04/19	13			█	█	█	█
66 Submit Final Report	16/04/19	16/04/19	1			█	█		
67 Submit Final Prototype	17/04/19	17/04/19	1			█	█		