Generating Life Stories From Facebook Posts

A Thesis
Presented to
the Faculty of the College of Computer Studies
De La Salle University-Manila

In Partial Fulfillment
of the Requirements for the Degree of
Bachelor of Science in Computer Science

by

HADE, Alden Luc R.
LAM, Janica Mae M.
SAAVEDRA, Camille Alexis T.R.
TE, Robee Khyra Mae J.

Ethel Chua Joy ONG
Adviser

July 13, 2017

Abstract

People love sharing stories with one another because stories provide them with enjoyable experiences as well as help them learn new things. Nowadays, people use social media, and in particular, Facebook, to share information about themselves, their daily activities and the things that interest them. However, a Facebook users data (posts, photos, personal info, etc.) by itself does not provide a *concise* narrative of events that can be used to adequately tell a persons life story.

There is no current work implemented which creates a textual story from a given set of user-generated social media data. This research focuses on automatically generating a life story from a person's Facebook text posts. To this end, a system called FB Stories was developed. It extracts user data from Facebook (with their permission), processes them, organizes them into different types according to their text content, and utilizes this knowledge in order to generate a persons life story.

The development of the system showed the difficulty of working with noisy user-generated data. The testing showed that, while there were issues with low precision and high recall, the system addresses the research gap and satisfies all of its objectives.

Keywords: Social media, Facebook, Text understanding, Post classification, Life event detection, Storytelling, Story generation

Contents

List of Figures

List of Tables

Chapter 1

Research Description

This chapter explains the concept of stories as an important part of human life. It introduces the current state of technology, and discusses the objectives of the research, the scope and limitations, and the significance of the study.

1.1 Overview of the Current State of Technology

The world is full of stories. In his book, *The Storytelling Animal*, Jonathan Gottschall says that human beings are natural storytellers – "that they can't help telling and *creating* stories because they like narratives so much" (Gopnik, 2012).

Storytelling is an ancient and abiding art. It is one of the things that distinguishes human beings from animals. The reasons why people love stories vary from person to person – *history* shows that for the longest time, humans loved to tell stories (Gopnik, 2012); *theologians* tell stories because their moral lessons influence readers to become better people; and *biology* claims that compared to when one simply explains things to someone, telling a story puts their whole brain to work (Widrich, 2012) – but in summary, people love sharing stories because they provide them with enjoyable experiences as well as help them learn. Stories also serve as a reflection of people's own experiences, and they are an effective route in reaching out and connecting to others.

The world of storytelling has changed over the course of history, from oral tradition to online technology. Nowadays, computers are still not capable of fully developing and telling stories on their own; recently however, more storytelling systems (also known as *story generation systems*) are being developed as part of

artificial intelligence (AI) trends to build solutions that could support and mimic human tasks.

Aside from storytelling systems, the digital age introduced society to the concept of social media, another method through which people's stories could be preserved. The introduction of social media has allowed storytelling to become more immersive, as it has employed a combination of text, photo, and video in a way that has become more participatory.

A *social networking site (SNS)* or social media is defined as "a web-based system that allows individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system" (Boyd & Ellison, 2010). The most popular SNS at the time of this writing is Facebook, which allows users to create their own profile with information about themselves; observe other users' content; and interact with others through reacting, commenting, and sharing. By far the biggest social network worldwide, Facebook has an estimate of over 1.71 billion monthly active users, as of 2016 (Harden, 2016).

Facebook is emerging as a near-universal storytelling method. One of the things that make Facebook successful is the unique, freeform nature in which it allows users to share information. Facebook's *timeline* feature provides people with their own way of creating a complete story about themselves, from their birth to the current day. Users, pages, groups, and events each have a timeline containing posts that they are involved in. Information on Facebook can consist of many forms: from text, to photo, to video. These small acts of posting and updating one's status about current events and occurrences on Facebook can be considered small stories as the posts are arranged chronologically similar to a storyline (West, 2013).

Recognizing the appeal and importance of remembering past events, Facebook has recently implemented a feature called *On This Day*, which notifies users of posts that happened on a particular day in the past several years. Facebook also has a feature called *Year in Review*, which collects photos from what Facebook determines to be the user's most significant moments in the past year. However, while Facebook already has features that can potentially tell stories about a person's life through pictures, there is no current work implemented which creates a textual story of a user's Facebook account. Research works in automated story generation have also not explored the use of user-created content (such as social media posts) as a source of knowledge for planning the content of a story to be generated.

1.2 Research Objectives

This section presents the general and specific objectives of the research in order to be able to address this research gap.

1.2.1 General Objective

To develop an application that generates one's story using data collected from his/her Facebook account.

1.2.2 Specific Objectives

1. To define a life story and determine its elements and structure;
2. To review the content of Facebook and the existing methods used in extracting data from Facebook;
3. To design an algorithm for understanding user-created text content;
4. To build a knowledge base from which to model the story structure and to supplement the data extracted from Facebook;
5. To select which post types and post classification algorithm to use in classifying each Facebook post;
6. To analyze story generation algorithms and design an algorithm for generating stories using data from Facebook; and
7. To define the metrics to be used in evaluating the generated story.

1.3 Scope and Limitations of the Research

A *life story*, or a biography, is an account of the series of events making up a person's life according to The Free Dictionary [1]. In order to apply this concept in the research, the different elements which make up a life story should be examined. However, the idea of a story is not to capture every single detail, but to give the

[1] The Free Dictionary. http://www.thefreedictionary.com/

participants something to remember. Therefore, the more important and relevant elements that make up a good story should be determined.

A Facebook user's account can contain gigabytes of unnecessary data, which are irrelevant in generating a complete story. Therefore, this research needs to determine which types of content are appropriate for generating life stories. Furthermore, knowledge of the different types of methods of extracting data from Facebook allows the research to extract the data needed in generating a story.

Stories are based on user-generated data; therefore, consent is necessary not only to provide the necessary data but also to avoid violating privacy and confidentiality. The system would be launched by the users themselves, and not by somebody else. Graphic files such as images and videos will not be included in this research. The number of Likes for each post are extracted but not the information about the actual users. Private messages are not included.

Story generation systems require certain predefined knowledge in order to do their task of generating stories. A knowledge base, which is defined as "the underlying set of facts, assumptions, and rules which a computer needs to solve a problem" according to Oxford Dictionary [2], helps the system generate the story. This research requires identifying possible knowledge sources as well as designing an appropriate knowledge base structure to provide data to be used for generating a life story.

Social networking sites have been a crowd-sourced knowledge base of people's activities as well as real time events (Jain, Kasiviswanathan, & Huang, 2016). Starbird and Palen (2011) stated that rescue workers use SNSs to know and talk to natural disaster survivors. In this research, for stories to have a better flow, it is important to classify posts accurately according to their types. However, there are many post types present in SNSs, and accommodating all these is not possible in this research. Thus, only the most used post types are selected. Different classification algorithms will also be reviewed to determine which ones can be adapted for this research.

Story generation algorithms (SGAs) are reviewed in order to determine which ones are appropriate and can be adapted for the context of this research. These are computational procedures which result in an artifact that can be considered a story (Gervás, 2012). The concept of a story in SGAs is functional and not aesthetic, which means that having appealing text is not their primary concern.

Finally, in order to measure the quality of the generated stories, a set of evaluation metrics are defined for this research. These metrics would then be

[2]Oxford Dictionary. https://en.oxforddictionaries.com

used to evaluate the strengths and weaknesses of the system.

1.4 Significance of the Research

As mentioned in Section 1.1, despite the appeal of storytelling as part of human experience, computers are still not capable of fully developing and telling stories on their own, nor understanding stories being told by humans. This research, therefore, contributes to the field of computing technology by contributing to the field of story generation: by enabling computers to make sense of a wide variety of user-generated data in order to tell stories.

This research can be a first step needed by a smart computer to understand a person's life. With this, software agents can use information from Facebook data to make sense of a person's activities and experiences. This can help inform relevant stakeholders regarding aspects of lives of their constituents, with applications in social behavior analysis, community healthcare monitoring, and personalized digital marketing.

This can lead to a better understanding of people, both as individuals and as a whole community, and open up possibilities of customization and personalization in computer-based support systems and interaction. Furthermore, this research can be of interest to writers and computer scientists looking to learn more about the fields of artificial intelligence and/or story generation.

1.5 Research Methodology

This section lists down and elaborates on the specific activities that were performed by the proponents over the course of conducting the research. The discussion includes activities done as well as ethical issues encountered.

1.5.1 Research Activities

1.5.1.1 Review of Related Literature

During this phase, existing works related to event classification, story generation systems, and text understanding were reviewed and analyzed. In addition, Face-

book and other relevant social networking sites were examined to determine which ones are most apt to provide enough data to be able to write a person's life story.

1.5.1.2 Review of Ethical Issues

In this phase, the ethical issues concerned with research were dealt with. The General Research Ethics Checklist (Appendix F) was accomplished to be able to identify potential risks to participants, as well as the Research Ethics Checklist for Investigations Involving Human Participants (Appendix G).

The data collection process, the sources of data, the sampling details, and the data retention details were provided. Two Informed Consent Forms (Appendix H and Appendix I) were created to be filled up by the users and were used during the data gathering process and the actual testing of the software.

Aside from accomplishing these forms, specific steps to address ethical concerns have also been identified. The steps are discussed in Section 1.5.1.6 Testing and Evaluation of this document.

1.5.1.3 Data Gathering and Analysis

To focus on analyzing the different elements that comprise a story, such as character, setting, and events, Facebook data from different users were gathered. From there, the scope was narrowed down to the elements that were deemed necessary, such as the subject's childhood, education, likes, and recent events in their life. The expertise of linguists and/or story writers have been consulted as necessary, as well as the knowledge gained from the Review of Related Literature.

1.5.1.4 Software Design

During this phase, the different modules of the story generation system have been designed. An overview of the system was documented in Chapter 4, FB Stories; and more detailed specifications were specified in Chapter 5, Design and Implementation.

With the help of the sample stories gathered from some Facebook users, mock stories were generated (manually) to get a better understanding of what sort of stories can be written from Facebook data. From these, story templates were created for use in parts of the output. Different approaches for post classification,

text understanding, and story generation have also been studied and applied as necessary, such as the concepts of post classification, Rhetorical Structure Theory (RST), and Resource Description Framework (RDF).

1.5.1.5 Software Implementation

The software was implemented based on the specifications in Chapter 4, FB Stories. Implementation and testing were done iteratively to focus on improving the system to achieve the target objectives and produce better stories. Essential steps to follow in software implementation were (1) building the knowledge base, (2) implementing the algorithms for data extraction, post classification, text understanding, and story generation, and basically following the design specifications in Chapters 4 and 5.

The software development life cycle followed was *Scrum*, an iterative and incremental agile methodology which splits the whole process into smaller tasks. This development process has series of iterations called *sprints* which lasts for no more than a month. This allowed easier adaptation to changes after performing unit testing to allow improvements on the current design.

1.5.1.6 Testing and Evaluation

The software was tested by Facebook users. Before the evaluation process, the metrics to be used by these evaluators were defined, based on Section 3.7 Evaluation Metrics. After this, the software underwent usability testing to determine the ease with which the user can his/her tasks. Different test cases were designed and executed, such as checking the generated stories of various Facebook accounts.

The Facebook users were briefed on the evaluation process by informing them of the following: (1) introduction of the research topic, (2) demonstration of how the software works, (3) signing of the informed consent, (4) confidentiality in storing the generated story from their Facebook accounts for further improvements, and (5) interview with the Facebook users for their experience in using the software as well as their feedback on the resulting stories. After this, the actual testing of the software was conducted in a room with internet connectivity.

Each session with a user lasted only for the duration of generating a minimum of one (1) story from the user's Facebook posts. After testing the software, the Facebook user was asked to answer a survey form to evaluate the correctness and completeness of the generated story. Qualitative feedback was also solicited, in-

cluding suggestions and recommendations to further improve the story generation.

Results from the testing process were used to determine the appropriateness and sufficiency of the knowledge sources and various data extracted from Facebook posts and their impact on the quality of the resulting stories.

1.5.1.7 Documentation

Documentation was done all throughout the duration of the research. The findings of each activity were documented accordingly.

1.5.2 Calendar of Activities

Table 1.1 shows a Gantt chart of the activities. Each bullet represents approximately one week's worth of activity.

Table 1.1: Timetable of Activities

	2016							2017								
Activities	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug
Review of Related Literature	●●●●	●●●●	●●●●	●●●	●●●●	●●●●									
Review of Ethical Issues		●●	●●●●												
Data Gathering and Analysis				●●●●	●●●●										
Software Design				●●●●	●●●●	●●	●●●								
Software Implementation							●●●	●●●●	●●●●	●●●●	●●●●	●●			
Testing and Evaluation									●●	●●		●●	●●		
Documentation	●●●●	●●●●	●●●●	●●●	●●●●	●●●●	●●	●●●	●●●●	●●●●	●●●●	●●●●	●●●●	●●●●	●●

Chapter 2

Review of Related Literature

This chapter discusses the features, capabilities, and limitations of existing research, as well as the algorithms and/or software that are relevant to the study.

2.1 Data Extraction Tools

Data extraction is the process of crawling through data sources, such as a database, in order to retrieve information that can be used for further data processing or data storage (Technopedia, 2016). For this study, data extraction involves selecting needed data from Facebook as an input to the story generation system.

Three data extraction tools that are relevant to the current study have been reviewed:

1. Download Facebook Data (Facebook, 2012; Pradhan, 2010)
2. Graph API (Weaver & Tarjan, 2013; Facebook, 2016)
3. Facepager (Keyling & Jnger, 2016)

Table 2.1 shows a comparison of these data extraction tools.

2.1.1 Download Facebook Data

The simple act of logging in to Facebook is already a way of accessing data stored in Facebook. The Activity Log is a built-in Facebook feature that shows the history of activities done on Facebook. It includes likes, comments, search history, and others. In 2010, Facebook introduced a new feature where these data can be downloaded for the user's own purposes. This is a compilation of the user's own data on Facebook, such as the user's personal info, friends list, and messages, among others.

Facebook automatically compiles the data to be downloaded once the user chooses to archive his/her files. An email containing the download link would then be forwarded to the mail address registered in the user's account. Once downloaded, the .zip file includes all media posted by the user in Facebook. It also contains a file, index.html, which is an HTML file that acts similarly to Facebook's home page. It displays the user's profile data such as the Facebook link, email address, registration date, birthday, past Facebook names (if any), current city and hometown, and other information that the user has provided on this Facebook account. There are also several HTML files provided in the HTML folder for easier indexing and browsing, as shown in Figure 2.1.

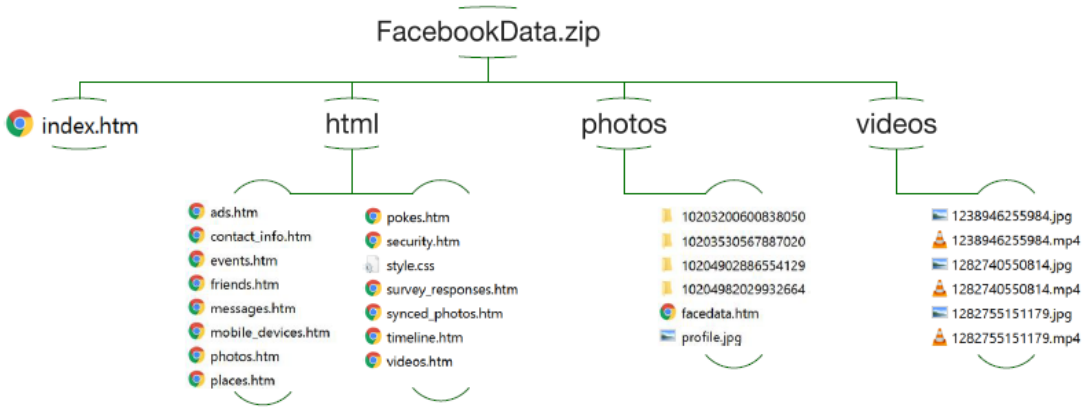

Figure 2.1: Content of Facebook Data .zip File

The date, time, timezone, tags, and source of posts that are shared are provided in the Timeline tab. Meanwhile, image and video files are separated from the description that comes along with it. Similar to the Timeline tab, the photos and videos tab also contains the time, date, and timezone of the media file posted. Aside from displaying the media file and the missing message of the post, the photos include the following details - the camera model, orientation, width, exposure, ISO speed, IP address and comments - while the videos include the thumbnail image and comments in the video.

2.1.2 Graph API

Facebook has developed the Graph API to allow developers to extract posts and other data from a specific Facebook page or a user's Facebook account. The API can be accessed from the URL, developer.facebook.com/tools, once a user has logged in.

Facebook's Graph API supports developers by supplying services such as providing snippets of codes for easier integration in JSON requests and responses. In Figure 2.2, GraphRequest calls "/user/me" to fetch the user data for the given access token. An access token controls data access and contains the permissions that the user has allowed, as seen in Figure 2.3. This access token expires after one hour. If no access token is provided, Graph API would only return publicly available information. The response data is deserialized into a JSONObject if the request is successful.

Figure 2.2: Available Coding Platforms and Code Snippets in Graph API (Facebook, 2016)

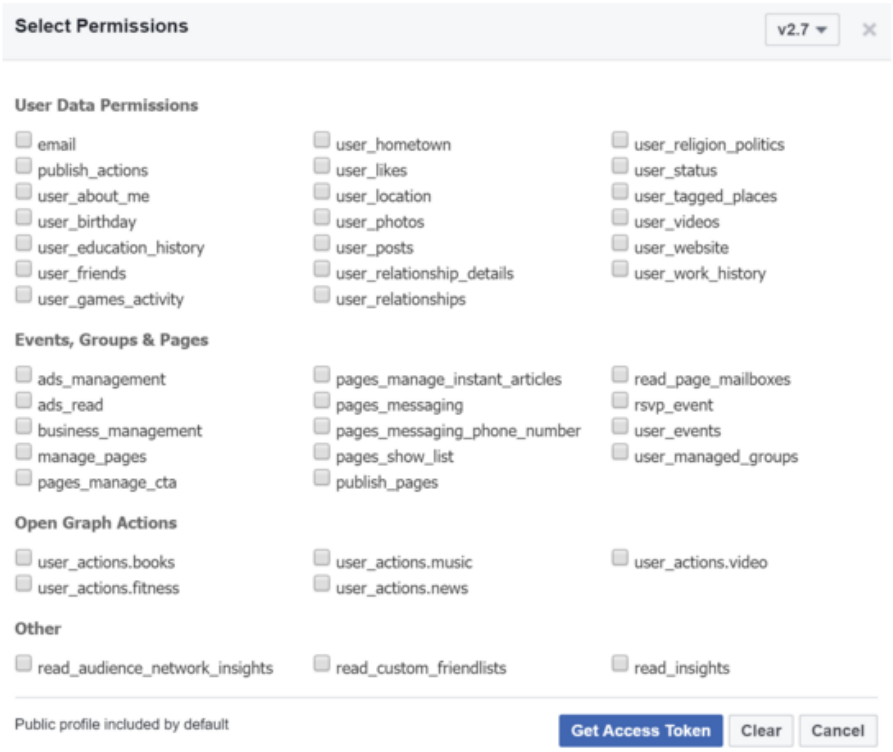

Figure 2.3: Permissions in an Access Token (Facebook, 2016)

2.1.3 Facepager

Facepager is a free and open-source tool that has gained quite a lot of interest for its simple process of collecting data through application program interfaces (APIs). Using Facepager does not require knowledge of programming because of its simple design. Presets are also available to help beginners and first time users when working with Facepager. Students and researchers commonly use this tool to gather data for projects such as for analyzing political communication on Facebook, and market analysts also make use of this tool to analyze data regarding a certain product.

Public data from Facebook, such as pages of companies or artists, enables Facepager to extract the content of the post, the number of likes, shares, and comments as well as other details provided.

Data are collected and extracted depending on the provided object ID or Facebook page name. All these collected data are then stored in a SQLite database that could later be exported to a Comma Separated Values (CSV) file for further analysis. Each row or tuple would contain the name of the page, message, type (indicated whether it is a video, link, photo), metadata type, likes, likes count, shares count, comments count, time it was created as well as the time the post was updated.

Table 2.1: Comparison among the different data extraction tools.

Data Extarction Tool	Data being extracted	Extraction of Data	Input	Output	Domain	Free	Platform
Download Facebook Data	Activity Log; User Profile Information; Messages; Friend List	Facebook User Account	N/A *Only requires user to log in*	Zip file containing html files and the media files	Facebook	Yes	HTML
Graph API	Activity Log; User Profile Information	Facebook Pages or Facebook User Account	Object ID and Access Token	JSON Object	Facebook	Yes	JavaScript PHP Android SDK iOS SDK cURL
Facepager	Facebook resources	Public Facebook Pages; Public Twitter pages; Public youtube accounts	Object ID	Data stored in SQLLite or CSV	Facebook Twitter Youtube	Yes	Python

2.2 Text Understanding Tools

Text understanding is the process of inferring the intended meanings of text made in natural language, such as greetings, commands, or messages. It consists of (1) reading texts that were formed in natural language; (2) determining the implicit and explicit meaning of each element, including words, phrases, sentences, and paragraphs; and (3) making inferences based on the implicit and explicit properties of these texts (Zhang & LeCun, 2015).

Text understanding tools are relevant in this research as a means of interpreting the textual contents extracted from Facebook. The text understanding tools reviewed in this study are the following:

1. FastText (Joulin, Grave, Bojanowski, & Mikolov, 2016; Mannes, 2016)
2. DeepText (Abdulkader, Lakshmiratan, & Zhang, 2016)
3. Google Cloud Natural Language API (Google, 2016)
4. Stanford CoreNLP (Manning et al., 2014)

Table 2.2 shows a comparison of these text understanding tools.

2.2.1 FastText

FastText is an open-source library for building text representation and classification (Joulin et al., 2016). It combines a number of language processing and machine learning concepts introduced in recent years, such as bag of n-grams and subword information. Different concepts are employed for two main purposes: efficient text classification, and learning word vector representations.

The bag of n-grams process is fast because it focuses more on the occurrences of a word as opposed to word order (Mannes, 2016). Words are represented in a multidimensional space, and linear algebra is used to calculate the relationship between a query and a categorized set of words. Via this method, a problem which is *qualitative* in nature– text analysis– becomes qualitative through the addition of statistics. Essentially, this enables fastText to be faster than traditional deep learning methods. Figure 2.4 shows a comparison between fastText and other text representation libraries.

FastText is not restricted to English and can work with other languages including German, Spanish, French, and Czech (Mannes, 2016).

	Yahoo		Amazon full		Amazon polarity	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
char-CNN	71.2	1 day	59.5	5 days	94.5	5 days
VDCNN	73.4	2h	63	7h	95.7	7h
fastText	72.3	5s	60.2	9s	94.6	10s

Figure 2.4: A comparison between fastText and deep learning-based methods.

2.2.2 DeepText

DeepText is a "deep learning-based text understanding engine." It can understand with near-human accuracy the textual content of several thousand posts per second (Abdulkader et al., 2016). It aims to help users make tasks, such as defining words or reserving plane tickets, easier.

Facebook users are composed of people with different cultures. Thus, it is important for DeepText to be able to interpret as many languages as possible. Another goal of DeepText is to better understand words in different languages using labeled data instead of traditional NLP techniques that would normally take longer processing time. In traditional NLP, the understanding part is highly dependent on stored knowledge thus requiring each word to be exactly the same for it to be understood. However, with deep learning, word embeddings could be used to get a sense of the word. According to Facebook, it is a "mathematical concept that preserves the semantic relationship among words". By mapping words and phrases into a common embedding space, it might know which words written in different languages yield the same meaning.

2.2.3 Google Cloud Natural Language API

The Google Cloud Natural Language API offers powerful text analysis by using machine learning models that are capable of revealing the structure and meaning of text. It can extract information about entities (such as people or places) which can be found in documents, articles, or blogs. Google uses the same machine learning technology to understand and answer user questions in a Google search query.

The Natural Language API features syntax analysis, entity recognition, multilanguage, and integrated REST API. Syntax analysis is the process of extracting tokens and sentences, identifying the different parts of speech, and creating depen-

dency parse trees for each sentence. Entity recognition seeks to locate and classify entities in text into predefined categories such as person, organization, location, events, and media. The API allows analysis of text in different languages, including English, Japanese, and Spanish. And lastly, the integrated REST API allows text to be uploaded in the request or integrated with Google Cloud Storage.

The API has several methods for performing analysis on text and each level of analysis provides valuable information for text understanding. One method is [1] syntactic analysis, which consists of the following operations: (1) sentence extraction, (2) tokenization, and (3) mapping. These will help in determining the syntactic meaning of tokens and their relationships. Syntactic analysis, as shown in Figure 2.5, is used for analyzing and parsing text.

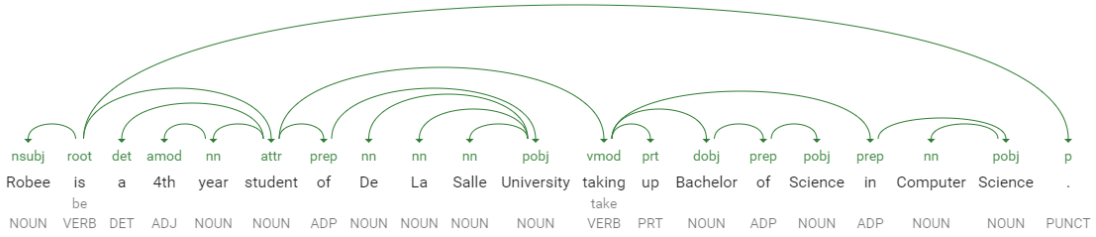

Figure 2.5: Sample output for syntax analysis

Another method is [2] entity analysis, as shown in Figure 2.6. It is useful for disambiguating similar entities and providing information about each entity found in the text. The analysis returns a set of detected entities, and parameters that are in connection with the identified entities, such as the type of the entity, its relevance to the text, and positions within the text where the identity was mentioned. Each entity is ordered from highest to lowest according to their salience scores, which reflects their importance in the overall text. Scores closer to 1.0 are highly salient, while scores closer to 0.0 are less salient.

2.2.4 Stanford CoreNLP

Stanford CoreNLP is a Java-based annotation pipeline framework which provides a set of natural language technology analysis tools ranging from tokenization to coreference resolution (Manning et al., 2014). It supports text in different languages such as Arabic, Chinese, English, French, German, and Spanish.

Stanford CoreNLP has several options to choose from when doing syntax analysis. But before the actual syntax analysis, it must first perform two operations: (1) sentence extraction and (2) tokenization. Sentence extraction is the process

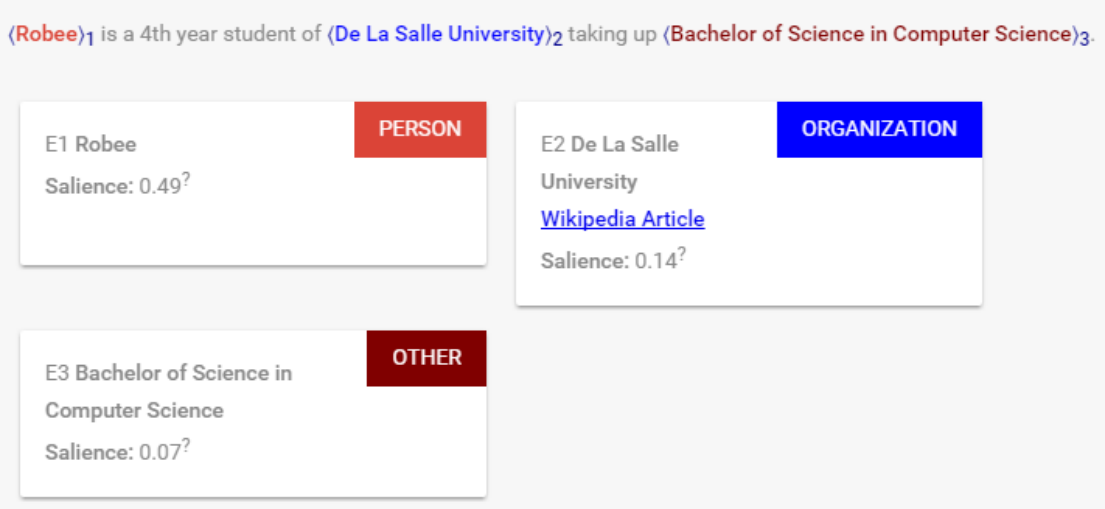

Figure 2.6: Sample output for entity recognition

of breaking down the text into separate sentences, while tokenization is the act of breaking down a sentence into tokens.

After that, the choices for syntax analysis include: (a) part-of-speech tagging (shown in Figure 2.7), (b) named entity recognition (shown in Figure 2.8), (c) lexical parsing (shown in Figure 2.9), or (d) universal dependencies (shown in Figure 2.10). Part-of-speech tagging returns the part of speech of each token; named entity recognition returns proper nouns; lexical parsing analyzes the grammatical structure of each sentence and provides a dependency tree; and the universal dependencies shows the relationship of a word to other words.

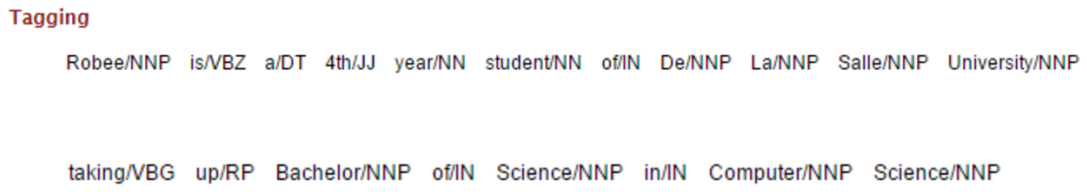

Figure 2.7: Sample Part-of-speech Tagging using Stanford CoreNLP

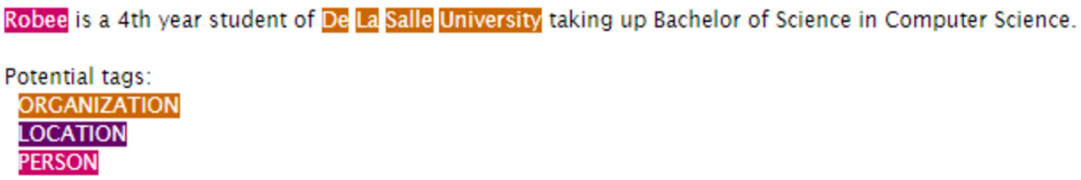

Figure 2.8: Sample Named Entity Recognition using Stanford CoreNLP

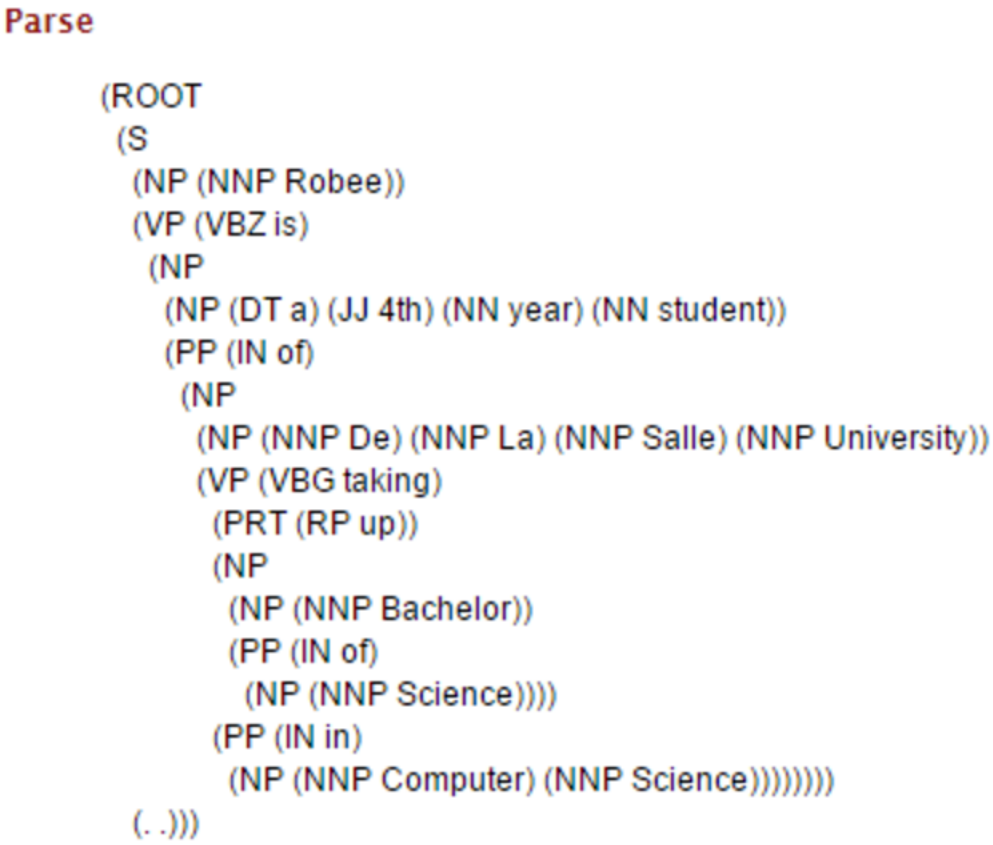

Figure 2.9: Parse Tree Output by Stanford CoreNLP

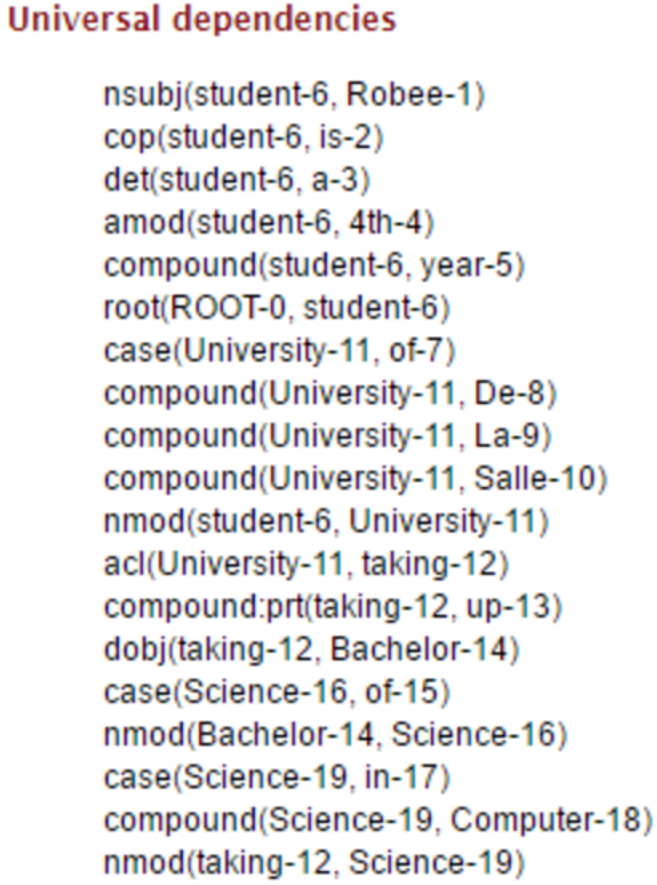

Figure 2.10: Universal Dependencies Output by Stanford CoreNLP

Table 2.2: Comparison of the text understanding tools.

Text Understanding Tool	Learning Process	Processes Available	Languages Supported
FastText	You have to train it yourself	Classification of posts	English, German, Spanish, French, and Czech
DeepText	Used FBLearner Flow and Torch for model training.	Entity recognition, General classification about the posts, extracting intent, and sentiment analysis	More than 20 languages
Google Cloud Natural Language API	Pre-trained	Entity recognition, Sentiment analysis, and Syntax analysis	English, Spanish, and Japanese
Stanford CoreNLP	Pre-trained	Arabic, Chinese, English, French, German, and Spanish	Part-of-Speech Tagger, Named Entity Recognition, Parser, Sentiment Analysis, Coreference, Universal Dependencies, Lemmatizer

2.3 Story Generation Systems

Story generation systems output a complete story of a certain genre given a (usually) unique set of inputs. The following story generation systems are reviewed in three aspects - the different inputs accepted and the output generated; and the process of planning the story:

1. Novel Writer System (Gervás, 2012, 2009; Méndez, Gervás, & León, 2014; Laclaustra, Ledesma, Méndez, & Gervás, 2014)

2. TALE-SPIN (Gervás, 2009; Mawhorter, 2013; Meehan, 1977)

3. Picture Books (Solis, Siy, Tabirao, & Ong, 2009; Ang & Ong, 2010; Adolfo, Lao, Rivera, Talens, & Ong, 2015)

4. Learning To Tell Tales (McIntyre & Lapata, 2009)

Table 2.3 shows a comparison among these story generation systems.

2.3.1 Novel Writer System (1973)

The first ever storytelling system was the Novel Writer System developed by Sheldon Klein. The system was recorded to have generated 2100-word murder-mystery stories in less than 19 seconds (Gervás, 2012). It uses a microsimulation model in which the behavior of each character and events were ruled by probabilistic rules that continuously changes the story (Gervás, 2009).

The user has to provide the character traits of the murderer/s and the victim/s with additional traits that describe their relationships with each other, their tendency to commit violence or sex, and the description of the setting in which the story will take place (Gervás, 2012; Méndez et al., 2014; Laclaustra et al., 2014). During the course of the story, some of the events causes motives to arise. Greed, anger, jealousy, and fear are possible motives for murder. The system can also generate more than one text-based story, indicating who the murderer is, why the murder was committed, and who discovered the murder incident.

The system generates stories based on two different algorithms: (1) a set of rules that indicates the possible changes from the current state of the story to the next and (2) a sequence of scenes corresponding to the type of story to be told. There are pre-defined set of rules to allow the construction of only one specific type of story.

2.3.2 TALE-SPIN (1977)

TALE-SPIN is a storytelling system that generates stories about the lives of woodland creatures developed by James Meehan (Gervás, 2009). It uses character simulation as a method in creating stories (Mawhorter, 2013).

The user chooses the traits and morals of the characters as well as the problem they will be facing (Gervás, 2009; Meehan, 1977). As the story progresses, new characters and items will be added as needed (Meehan, 1977). Relationships between the character may also arise to competition, dominance, familiarity, among

others (Gervás, 2009). The system then generates a story containing the events that are to happen in order to solve the problem.

The story has three main components: (1) the problem solver, (2) the goal, and (3) the sub-goals and actual events (Meehan, 1977). TALE-SPIN generates stories by combining both backward and forward chaining (Meehan, 1977). Forward chaining is expounding on the available data to be able to extract more data until the end goal is obtained, while backward chaining is described as working backward from the given goal/s, tracking down events that will yield the solution to the problem.

2.3.3 Picture Books (2008-2015)

Picture Books 1-4 is a series of existing automated story generation systems which use picture elements (setting, characters and items) as inputs from children in producing a fable-like text-based story. Stories are generated using Natural Language Generation (NLG) techniques. The stories to be generated follow the plot structure of exposition, rising action, climax and resolution.

The architectural design of Picture Books 1 is composed of three modules: the picture editor, the story planner and the sentence generator. The picture editor starts after the user chooses the setting of the story and the selection of characters and items come right after choosing the setting. The story planner module is subdivided into two modules: the story content planner and the story organizer. The story content planner chooses a theme based on the setting and items. The story organizer arranges and organizes the events of the story in an organized manner for the user. The result of this will be a story tree that will be used in the next module. The sentence generator module is again subdivided into two modules: the sentence planner and realizer. The sentence planner converts the story tree from the story organizer to sentence templates. The realizer then uses the SimpleNLG realiser which is an open source Java class library in converting the sentence templates given by the sentence planner to the actual sentences that made up the story. The final output of the realizer includes the story title and the story itself.

Picture Books 2 targets a slightly older age group comprising of children aged six to eight years old (Ang, Antonio, Sanchez, Yu, & Ong, 2010). It creates a more creative environment through depicting stories in a sequence of scenes through the given images and characters provided. The new environment required new semantic relations to be added to define the concepts dealing with movements and positioning of the objects in the story. It is capable of develop-

ing three or more scenes portraying various scenes in the intended story where in the characters exhibit three traits assigned by the system which helps depicts the story to the children. The system structure is similar to Picture Books 1 starting with the story editor module which initializes the environment for the user. The story planner module experienced changes as it aims to impart a moral lesson to the user with the use of the character's actions. It is comprised of three modules, mainly the theme formulator, setting formulator, and event generator. The theme formulator identifies the background, character and objects present in the scene and formulates a theme defining the character trait that it lacks for further development. The setting formulator focuses on describing the location and time the story takes place which is identified through a grid placed in the scene. Events generation involves the conceptual relations on the specified theme. Lastly, the sentence planner transforms the sequence of conceptual relations using character goals into readable sentences which involves aggregation, lexicalization and referencing.

In Picture Books 4, the system automatically generates stories based on pictures or stickers the children puts on the scene (Adolfo et al., 2015). Similar to Picture Books 2, it focuses on children from six to eight years old. It has the capability of generating stories consisting of single to multiple scenes, and can generate stories with a single character or with multiple characters, however, it focuses more on developing stories with multiple characters by incorporating interaction between them. Unlike Picture Books 1 and 2, which only caters one type of interaction, character-to-object, and character-to-character interaction, respectively, Picture Books 4 caters to both types of interaction and two additional types of interaction which are the verbal interaction and non-verbal interaction. The system uses both character-centric and author-centric approaches. The character-centric approach allows characters to have their own individual goals and pursue these goals in a way that is consistent with their personality and desires, while the author-centric approach makes sure that the entirety of the story revolves around the main topic. The system has three modules that were adapted from previous Picture Book systems.

2.3.4 Learning To Tell Tales (2009)

Learning To Tell Tales is a data-driven approach in storytelling (McIntyre & Lapata, 2009). The user provides the topic that may come in the form of phrase or sentence and the desired length of the story. The output is a text-based story generated after looking in the knowledge base containing the topic.

The system conceptualizes the story generation process as a tree after con-

sulting the knowledge base. The story tree Figure 2.11 has different levels which represent the number of sentences to be generated in the output. Additionally, each sentence in the tree has their own score. To generate a story, the system traverses the tree and chooses the node with the highest score. The system also uses two different searching methods: (1) searching for the best story and (2) searching for the most suitable sentences that can be gathered from the knowledge base.

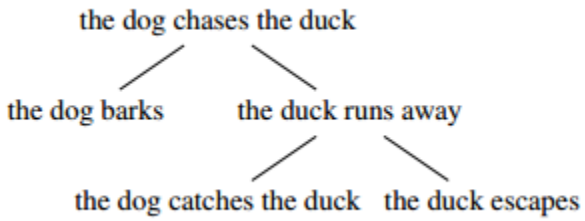

Figure 2.11: Example Story Tree
(McIntyre & Lapata, 2009)

There are four modules in order to generate a story: (1) content planning, (2) sentence planning, (3) surface realization, and (4) story ranking. First, all the possible verb-subject, verb-object, verb-adverb, and noun-adjective relations related to the topics are extracted from the corpora stored in the knowledge base and each of these verbs will have a score. Second, the system must combine each of them to form a sentence. The grammar rules will act as a template. Third, the sentences will undergo surface realization which fixes the grammar of each sentence. Lastly, the system performs a story ranking to know whether the story generated is interesting and coherent.

Table 2.3: Comparison among the different story generation systems.

Story Generation System	Genre	Target	Theme	Input	Output	Approach	Goal
Novel Writer System	Murder	Not specified	Weekend party	Murderer/s Victim/s Description of Setting	Text-based story	Rule-based	No
TALE-SPIN	Aesop's Fable-like stories	Not specified	Not specified	Initial Setting Traits or morals of characters Goal	Text-based story	Not Specified	Yes
Picture Books	Fable	Children (age 4 to 6)	Not specified	Settings Character Items	Text-based story	Author-centric Character-centric	No
Learning To Tell Tales	Not specified	Young Children	Depends on the gathered information	Topic Length of Story	Text-based story (based on the length specified by the user)	Knowledge-based reasoning	No

2.4 Social Networking Sites

Social networking sites (SNS) are popular tools for social interaction as well as information exchange between individuals (Hughes, Rowe, Batey, & Lee, 2012). They serve as virtual communities which allow people to connect and interact with each other on a particular subject, or to just "hang out" together online (Cheung, Chiu, & Lee, 2011).

The social networking sites reviewed here are Facebook and Twitter.

2.4.1 Twitter

Twitter is commonly referred as a micro-blogging website. This is a combination of instant messaging and blogging that allows users to post or send a short message, but is limited to 140 characters, called "tweets" (Crymble, 2010). In 2010, Twitter's population grew to 175 million registered users from 30 million users early that year (Rao, 2010). Registered users can read and post tweets, retweet other's tweet, and follow other users or industry figures for updates, while unregistered users can only read tweets from public accounts.

Twitter has become popular in today's society for several reasons. It is one of the first sites to introduce the *following* and *followers* concepts that appeal to the audience as they can easily follow or unfollow other users. It is also one of the most simple social media sites to use because of its simplicity and easily navigable interface. In addition, Twitter allows users to easily share information.

However, Twitter has some downsides. First, there is no system of accuracy as users can just say about anything. Second, users are limited to 140 characters, which means, they may not be able to explain themselves or their thoughts in detail. Many tweets involve either a use of "(con't.)" and/or a reply to the own tweet to signify a chain of tweets that are altogether expressing a single thought. And lastly, Pear Analytics, a firm that specializes in marketing analytics, conducted a study that categorized tweets into 6 areas - News, Spam, Self-Promotion, Pointless Babble, Conversational and Pass-Along Value. They found out that 40.55% of the tweets in the gathered sample fit into the category of Pointless Babble, followed by a 37.55% in conversational (PearAnalytics, 2009).

2.4.2 Facebook

Facebook is the most active SNS today. It consists of multiple avenues of social interactions and billions of page views, with more than 21 million users as of 2007 (Ellison, Steinfield, & Lampe, 2007) and has dramatically increased to almost one billion users worldwide as of 2013 (Farahbakhsh, Han, Cuevas, & Crespi, 2013). The comScore Media Metrix Multi-Platform conducted a study in 2015 showing a huge gap between other social networks and Facebook for being the most accessed site for millennials as shown in Figure 2.12 below. According to Dave Chaffey (2016), Facebook continues to dominate the social landscape, leading number one on social media platforms around the world.

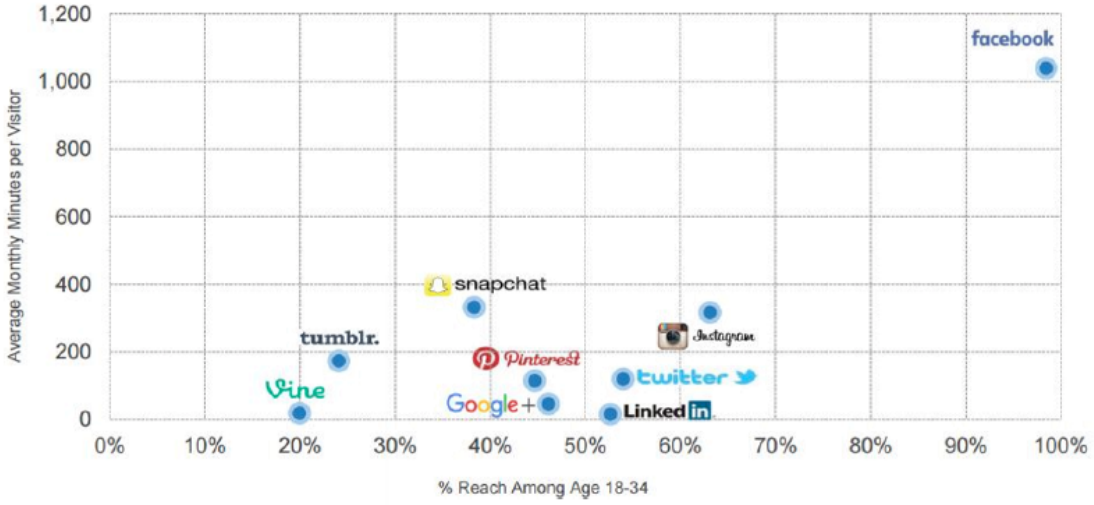

Figure 2.12: Millennial Demographics in Social Media Platforms (Chaffey, 2016)

Facebook is a convenient and open social media platform that welcomes not just individuals but also companies and organizations to communicate and interact with one another. Westlake (2008) states that "Facebook develops technologies that facilitate the spread of information through social networks allowing people to share information online the same way they do it in the real world". It promotes interactions between users, through simple status updates, posts, or shares, and enables users to inform others about their whereabouts and actions.

In Facebook, users are required to create visible profiles providing their name, gender, date of birth and email address. In addition to that, more information such as contact information, personal interests, background information and favorites (books, movies and music) can also be added to the profile at the discretion of the user (Nadkarni & Hofmann, 2012). Posting information about their interest

enables others to get a grasp of one's characteristics or personality (Farahbakhsh et al., 2013). Derek Coles (2015) said that "Facebook is the modern way of not being on your own" as it also allows you to interact and meet people you don't know and enables people to discuss topics with common interests.

A study by Ashwini Nadkami and Stefan G. Hofmann (2012), showed that people tend to stay online and use Facebook because posting in Facebook allows self-promotion and influences one's self-esteem and self-worth. Self-esteem and self-worth are affiliated with the sense of belonging in the society or a acceptability to a group, which, according to Nadkami & Hoffman, can be measured by the number of likes and comments received on one's share or post or a simple act like being tagged in a group picture. The study concluded that "Facebook creates an environment where information is shared proactively because of the site's influence on a user's need for popularity" (Nadkarni & Hofmann, 2012).

2.5 Knowledge Base

According to the American Heritage Dictionary (2011), a *knowledge* base is a collection of data organized in a form that facilitates analysis by automated deductive processes. It can be perceived as an expert system (Mars, 1995). It uses a *knowledge representation language* for expressing rules and/or objects.

The different knowledge base systems reviewed in this section are as follows:

- Cyc [1] (*http://psych.utoronto.ca/users/reingold/courses/ai/cyc.html*, 1999; *Overview of Cyc Inferencing*, 1994);
- WordNet [2]; and
- ConceptNet (Liu & Singh, 2004).

Table 2.5 shows a comparison among these knowledge base systems.

2.5.1 Cyc

Cyc is an artificial intelligence project that aims to collate and assemble a complete knowledge base of everyday common sense knowledge that humans inherently have

[1] OpenCyc. http://www.opencyc.org/
[2] WordNet. https://wordnet.princeton.edu/

[3]. This project is undertaken with the goal of enabling applications that use AI to perform reasoning that can be considered humanlike.

Cyc's knowledge base consists of data from news and magazine articles, encyclopedia entries, advertisements, and more. Cyc's knowledge base is represented as a directed graph containing the following:

1. Constants - terms that people understand;
2. Variables - case-sensitive unique identifiers;
3. Predicates - terms to represent relation types;
4. Formulas - expressions;
5. Logical connectors - and, or, not, implies;
6. Quantifiers - forAll, thereExists;

Figure 2.13 shows an excerpt of Cyc's knowledge base. Fido (variable) is a (predicate) dog (constant), for example.

Parts of the project were released to the public as OpenCyc, which provides an API, RDF redpoint, and data dump, all open-source [4]. As of OpenCyc 2.0, the knowledge base contains 239,000 concepts and 2,093,000 facts, and can be browsed on the OpenCyc website. The CycL and SubL interpreter (the program that allows you to browse and edit the database as well as to draw inferences) is released free of charge, but only as a binary, without source code. It is available for Linux and Microsoft Windows.

Cyc also contains an inference engine – that is, a computer program that derives answers from a knowledge base [5]. The Cyc inference performs general logical deduction, used for reasoning.

2.5.2 WordNet

WordNet [6] is a large lexical database of English words. A lexical database is a database of words and information about those words; in other words, a lexical

[3]Cyc. http://psych.utoronto.ca/users/reingold/courses/ai/cyc.html
[4]OpenCyc. http://www.opencyc.org/
[5]Cycorop. http://www.cyc.com/overview-cyc-inferencing/
[6]WordNet. https://wordnet.princeton.edu/

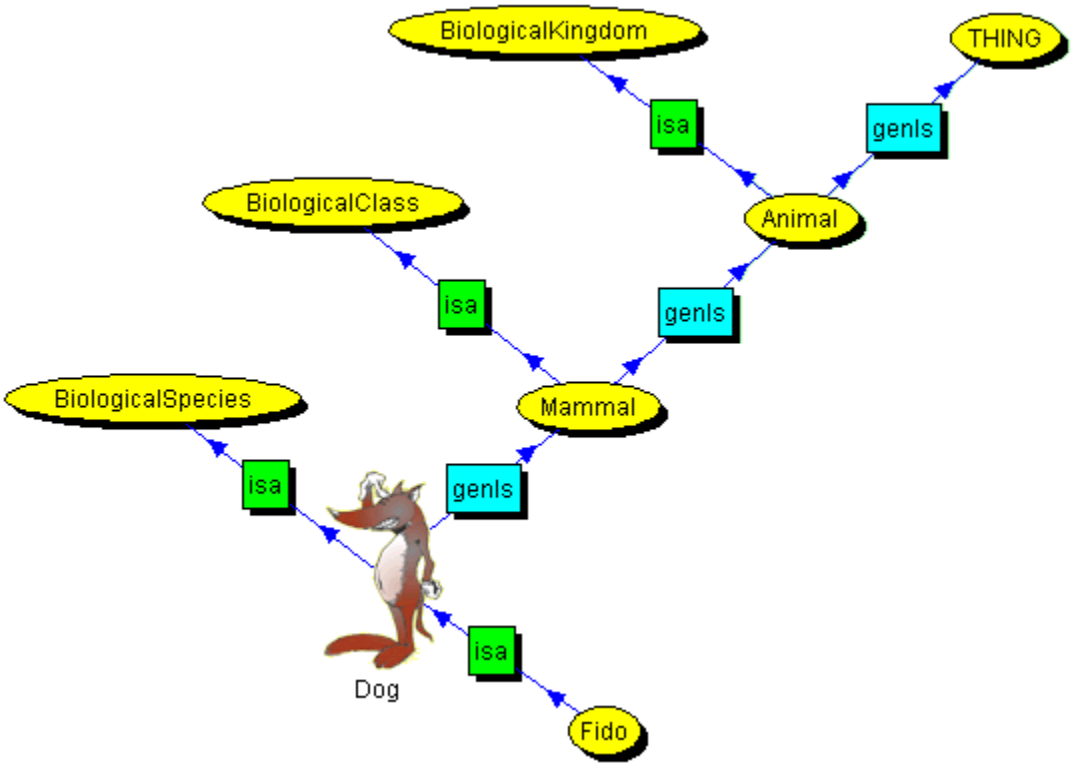

Figure 2.13: An excerpt of Cyc's knowledge base, showing common sense knowledge about a dog named Fido.

database is a dictionary that is easily machine-readable according to The Free Dictionary [7]. In particular, WordNet is a database of words, primarily nouns, verbs, and adjectives, grouped into synonym sets, or "synsets". Rather than simply words, however, WordNet contains specific "senses" of words (a "sense" is a distinct meaning that a word can assume). These senses are linked to each other by semantic relations such as synonymy and "is-a" hierarchical relations (e.g. "dog is a mammal").

As of WordNet 2.0, the database contains around 200,000 word senses (Liu & Singh, 2004). It is praised for being easy to use. As a simple semantic network with words at the nodes, it can be readily applied to textual input for purposes of getting more relevant results from a simple query, or determining semantic similarity between words.

[7] The Free Dictionary. http://www.thefreedictionary.com/

2.5.3 ConceptNet

ConceptNet is a free common sense knowledge base and NLP toolkit developed by Hugo Liu and Push Singh (2004). Its information is based from the Open Mind Common Sense (OMCS) database. OMCS is an artificial intelligence project based on the Massachusetts Institute of Technology (MIT) Media Lab, and its goal is to create and maintain a large common sense knowledge base from the contributions of many people. In comparison to Cyc and WordNet, therefore, ConceptNet's knowledge base was filled in most part by the general public, rather than hiring a lot of knowledge engineers.

Common sense knowledge is knowledge that is inherent in most humans, enabling us to, for example, determine that a lemon is sour, a doorknob needs to be turned in order to open a door, or even why the sentence, "I got fired from my job today," contains a negative connotation.

ConceptNet's semantic network is expressed as a directed graph whose nodes are concepts which have atomic meaning (such as words like "chair", or phrases such as "wake up"). These nodes are connected to each other by common sense knowledge about them. For example, "waking up in the morning" includes "yawning" and "checking messages" as its subevents as shown in Figure 2.14

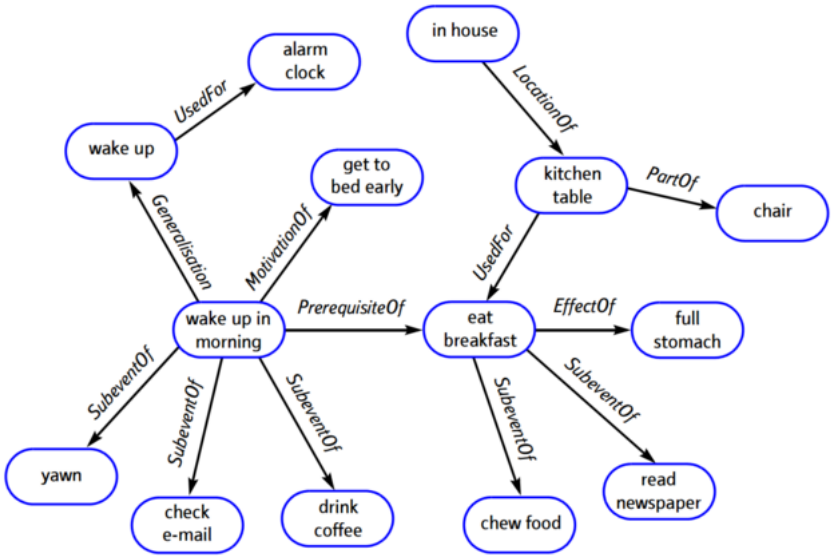

Figure 2.14: An example of ConceptNet's semantic network of knowledge. Concepts consist of a noun phrase along with an optional verb or prepositional phrase.

As of ConceptNet 5.0, there are over 3.9 million concepts, connected by 8.7 million assertions, such as "IsA" ("dolphin is a mammal"), "LocationOf" ("house

is the location of kitchen table"), or "SubeventOf" ("chewing food is a subevent of eating breakfast").

Through ConceptNet, it is easier to extract practical knowledge from text. This includes distinguishing between the correct definition of the word used in a context ("He ate chips with his lunch"), determining analogies, summarizing a topic, and event prediction.

Table 2.4: Comparison among the different knowledge base systems.

Knowledge Base	Knowledge Objects	Relations	Content	Source of Knowledge
Cyc	words	any	common sense data	magazine, news articles, encyclopedia, etc.
WordNet	words	word relations like hyperonymy, hyponymy, ISA relation, antonym, etc.	English words and information about these words	
ConceptNet	concepts (words or phrases)	semantic relations like LocationOf, IsA, CapableOf, SubeventOf, EffectOf, UsedFor, DesiredFor, etc.	Common sense knowledge	Open Mind Common Sense, DBPedia, Wiktionary, WordNet, OpenCyc, Verbosity, ReVerb, GlobalMind Translation, nadya.jp

2.6 Post Classification

A social media user's posts by themselves cannot provide a concise narrative of events in order to tell a complete life story. Story generators can be designed to utilize these events extracted from posts that users share about themselves into a life story. Before this can be achieved, however, the story generator needs to be able to classify posts based on their textual content.

With the volume of data available on social media platforms, NLP researchers have worked on putting some structure to organize text-based data to provide a more appealing interface (Setty, Jadi, Shaikh, Mattikalli, & Mudenagudi, 2014); to discover themes in disaster-related tweets (Syliongka et al., 2015); and to find patterns and glean community sentiments in election tweets (Wang, Can, Kazemzadeh, Bar, & Narayanan, 2012).

Table 2.5 shows a comparison among these post classification algorithms.

(Kinsella, Passant, & Breslin, 2011) state that social media, because of its informal and brief nature, presents a unique challenge for topic classification. They note that in social media, there is a frequent reliance on hyperlinks to external sites to give context to a conversation. The paper investigated the usefulness of metadata such as those hyperlinks in order to better understand the topic of a particular post. It concluded that including object metadata at all, not necessarily hyperlink metadata, outperforms classification which is based solely on the post's original text content.

The work of (Setty et al., 2014) involves dynamically classifying a Facebook user's news feeds into categories such as life events, entertainment and liked pages as a "better representation of data on the user's wall". The life event posts were further classified based on their sentiments as happy, neutral and bad feelings. These studies, however, have focused on finding patterns and trends from data coming from posts and tweets of multiple users over a certain period of time.

Choudhury and Alani (2014a) further noted that most research works in detecting events from social media content have focused on world events such as earthquakes and elections, and entertainment news. Focusing their own efforts on individuals, they detect common personal life events from Twitter to identify those that are interesting and important and can therefore be used to form part of a personal digital story book. They identified five events in a person's life as being the most important: [1] graduation, [2] marriage, [3] new job, [4] having a newborn child in the family, and [5] undergoing surgery. They also used related words to help find tweets that correspond to those events.

Table 2.5: Comparison among the different works regarding post classification and life story detection

	Focus	Social Network	Approach	Results
Kinsella, et al. (2011)	Posts with hyperlinks	Facebook	Multinomial Naive Bayes	F-score: 84% without hyperlinks; 90% with hyperlinks
Setty, et al. (2014)	Classifying posts on news feed	Facebook	SVM, Logistic Regression	Accuracy: 93%; Precision: 94%; Recall: 94%
Choudhury & Alani (2014a)	Detecting life events in posts	Twitter	Naive Bayes, Multinomial Naive Bayes, SVM	AUC: 77%; Precision: 80%; Recall: 85%

Chapter 3

Theorerical Framework

This chapter introduces and expands on the relevant concepts and theories to be used over the course of this research.

3.1 Life Stories

A *life story*, or a biography, is an account of the series of events making up a person's life according to The Free Dictionary [1]. An *autobiography*, or a *memoir*, is an account of a person's life written by themselves according to Oxford Dictionary [2]. An *event* is anything that happens, especially one of importance according to Oxfor Dictionary [3].

The following sections describe the different elements of a story, and the linguistic concerns surrounding the expression of these elements.

3.1.1 Elements and Structure of a Story

A story has a beginning, middle, and end portion (Pacis, personal communication, October 12, 2016). Failure to provide any of these parts renders a story incomplete and disinteresting. The linear structure of a story, as shown in Figure 3.1, is taken from (Pacis & Gojo-Cruz, as cited by (Chua, Cu, Ibarrientos, & Paguilinan, 2016))

[1]The Free Dictionary. http://www.thefreedictionary.com/
[2]Oxford Dictionary. https://en.oxforddictionaries.com
[3]Oxford Dictionary. https://en.oxforddictionaries.com

as well as from (Hancock, 1994 as cited in Types of Plot, 2010), who states that a plot is a "sequence of events that occurs to characters in situations in the beginning, middle, and end of a story.".

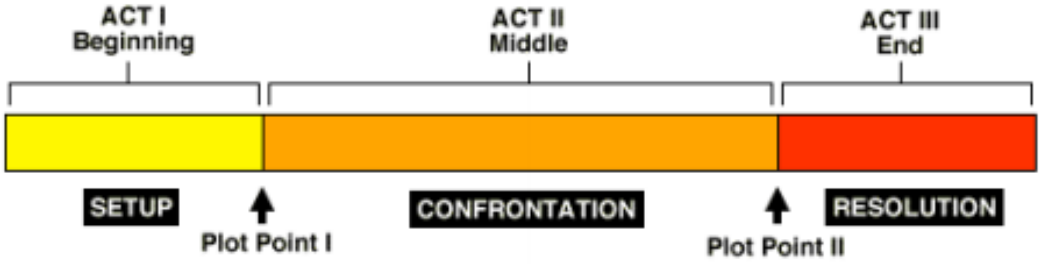

Figure 3.1: A model of the linear structure of a story.

As seen in Figure 3.1, events, details, or plot points, may occur that separate the beginning from the middle, and the middle from the end.

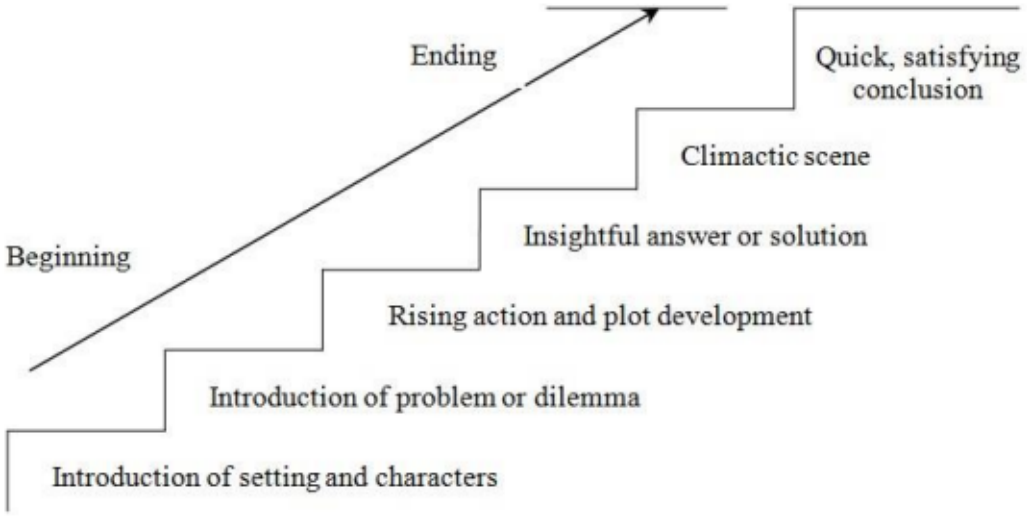

Figure 3.2: The story structure used in Picture Books, which tells the (fictional) story of a child who is disobedient.

Stories for children, such as those generated by the Picture Books story generation system (Solis et al., 2009), follow the structure of Machado (2003, as cited in (Solis et al., 2009)) which consists of the introduction; a problem; rising action; solution; and climax. Figure 3.2 illustrates the structure of Picture Books 1's stories.

Some of the elements present in Picture Books' stories are only necessary in the case of fictional stories, while some will still be necessary, such as having

a beginning and an end. Life stories, which are a different type of story from children's stories or other fictional stories, follow a different pattern. According to Youse (2005), there are ten (10) necessary elements present in a life story. These are:

1. **Birthday and birthplace**. The story should be able to indicate when and where a person's life started.

2. **Family members**. Was this person the eldest in a few children? Did he/she eventually take a spouse and sire any children? Did he/she have notable relatives?

3. **Childhood and education life**. Early achievements or stories that happened early on in this person's life may have had an impact later on in his/her life.

4. **Hobbies, interests, and notable activities**. This is present for the reader to be able to get an initial impression of this person. Do the person's hobbies or activities make them more interesting? Do they relate to other aspects of their life? Being able to craft interesting stories is important in this research, and describing a person's hobbies is one part of it.

5. **Photos or likenesses**. Providing a likeness of the person will complete the impression the reader has about that person.

6. **Anecdotes**. Interesting stories about this person as told by others, that makes this person more interesting.

7. **Career** (if the person is old enough to have had one). Is their work a big reason on why a story is being told about them? Does their career relate to their interests, or past experiences? Did they make significant contributions to mankind by doing their job?

8. **Reason for fame**. At what point in their life did this person become noteworthy or famous, and why?

9. **Later life** (if the person is deceased). Did they continue their work and/or contributions to mankind later on in their life? Were they honored for their achievements?

10. **Death** (if the person is deceased). When and where did they die, and under what circumstance? Was there anything unusual or significant about their death? For example, U.S. Presidents Thomas Jefferson and John Adams both died on 4 July 1826; 4 July being the American Independence Day.

All, or most of, these elements should be present in a life story. Such a story structure can be found by perusing articles about a person's life (e.g. Miriam Defensor Santiago). However, some elements (such as numbers 7-10 in the list above) are not possible to put into a person's life story because they may not be old enough. For these ones, concern should be put into ensuring that the story ends in a satisfying manner despite the missing details.

3.2 Facebook

3.2.1 Facebook Content

Facebook contains a lot of data ranging from texts to photos and to videos. It contains numerous of varying stories, facts and events from users all around the world. Each user has the ability to post any content they prefer to which can be categorized into 13 categories, as shown in Figure 3.3.

Figure 3.3: Facebook content categorized into 13 categories.

3.2.2 Facebook Components

Facebook is comprised of eight different components namely: [1] News Feed, [2] Friends, [3] Timeline, [4] Likes, Comments, and Shares, [5] Messages and Inbox, [6] Notifications, [7] Groups and [8] Statuses or Posts. In this section, only the status or posts component will be discussed since these will be the focus of the research.

3.2.2.1 Status or Posts

In 2013, Facebook has added a new structured status update feature allowing users to specify what they are feeling, watching, listening, playing, celebrating, among others (Darwell, n.d.). Those which we will be using for event classification in this research are the following:

a. Celebrating - Shows what the user is remembering at the time of posting.

b. Travelling To - Tells where the user is travelling at the time of the status update. The user can include the Facebook page of the location he/she is currently travelling to or he/she can simply include the location if there is no Facebook page for that.

c. Eating - Tells what food the user is consuming.

d. Drinking - Tells what beverage the user is drinking.

3.3 Data Extraction Tool

For this research, Graph API was used for data extraction. Graph API is a low-level HTTP-based API that is primarily used to access data and information in Facebook's platform (Facebook, 2016). It allows applications to do certain actions in Facebook such as publish updates, media files and even schedule a post.

3.3.1 Structure

The structure of Graph API consists of three main types, namely: nodes, edges and fields. Nodes are the basic components of Facebook such as a user, page, or comment; edges are the connections between the nodes, such as the connection between a user's photo and that photo's comment; and fields are the details or information about a specific node, such as the first and last names of the user.

These nodes can be accessed through making HTTP GET requests passed to the API at graph.facebook.com or graph-video.facebook.com for video uploads. Most APIs would require access tokens which determine the permissions for secure access to Facebook APIs. Each access token contains information about the token's expiration as well as the application in which it was generated (in this case, our system).

When a user logs into Facebook through an application (or app), the app will be able to obtain an access token which allows access to the user's data on Facebook (Facebook, 2013). Logging in only accesses the basic permissions such as the public default. Additional permissions can be listed down in the scope parameter and would inform whether the user would choose to authenticate the app with the said permissions (Facebook, n.d.-a). The user access tokens are then automatically stored by the Facebook SDK for JavaScript. These can be retrieved through FB.getAuthResponse which obtains the access token within the received response.

3.3.2 JSON File

Once the access token with the prefered permissions is obtained, the system can access the user's Facebook data through HTTP GET requests and receive a JSON file containing the requested information from the user. The structure of the JSON file received may vary depending on the node or edges read (Facebook, 2016). The general form for this is: { "fieldname" : {field-value}, . }

3.4 Text Understanding

For this research, the Stanford CoreNLP API, a service that gives access to a set of natural language analysis tools, was used. More specifically, the API's part-of-speech tagger, named entity recognizer, parser, and universerval dependencies were used to understand text. The API takes in a string as an input and its output can be viewed in four different ways, each of which was shown by the aforementioned tools.

Given the sentence, "Robee is a 4th year student of De La Salle University taking up Bachelor of Science in Computer Science.", the API produces the results shown in Figure 3.7, Figure 3.9, Figure 3.10, and Figure 3.11, for named entity recognition, part-of-speech tagging, parsing, and universal dependencies.

3.4.1 Named Entity Recognition

Named Entity Recognition (NER) is a process that classifies entities in the given text and categorizes them as person, organization, location, among other classifiers. The Stanford CoreNLP API determines the known entities and returns

information about those entities.

In the CoreNLP package, it has two classes, Annotation, and Annotator. Annotations are data structures that hold the results put out by Annotators. They can hold parsed data, part-of-speech tags, or named entity tags. Annotators, on the other hand, work like functions. They can parse and tokenize text and perform NER tagging on sentences. Annotations and Annotators are integrated by AnnotationPipelines, which can create sequences of Annotators.

To construct a Stanford CoreNLP object from a given set of properties, StanfordCoreNLP (Properties props) must be used. The method creates a pipeline using the given annotators in the annotators property.

```
Properties props = new Properties();
props.setProperty("annotators", "tokenize, ssplit, pos, lemma, ner, parse, dcoref");
StanfordCoreNLP pipeline = new StanfordCoreNLP(props);
```

Figure 3.4: Sample Code for Creating Pipeline

The code snippet shown in Figure 3.4 creates a Stanford CoreNLP object with the following annotators: named entity recognition, part-of-speech tagging, parsing, and lemmatization. For named entity recognition, the ner annotator is used.

```
String text = ... // Add your text here!
Annotation document = new Annotation(text);

// run all Annotators on this text
pipeline.annotate(document);
```

Figure 3.5: Sample Code for Parsing Text

After creating a Stanford CoreNLP object, the annotate(Annotation document) method is used to parse arbitrary text as shown in Figure 3.5.

The output of the Annotators is accessed using the data structures CoreMap and CoreLabel (in Figure 3.6). Entity type for each token can be retrieved using the get (NamedEntityTagAnnotation.class).

The analysis of the online version of Stanford CoreNLP from the sample input returned a set of detected entities namely, “Robee” and “De La Salle University”;

```
List<CoreMap> sentences = document.get(SentencesAnnotation.class);

for(CoreMap sentence: sentences) {

  for (CoreLabel token: sentence.get(TokensAnnotation.class)) {

    String ne = token.get(NamedEntityTagAnnotation.class);
  }

}
```

Figure 3.6: Sample Code for Named Entity Recognition

along with their corresponding types (shown in Figure 3.7).

Figure 3.7: Actual Named Entity Recognition Output by Stanford CoreNLP

3.4.2 Part-of-Speech Tagging

A Part-of-Speech Tagger is a piece of software that can read text in some language, break it down into tokens, and assign a part to each token, such as *noun*, *verb*, or *adjective*.

For the part-of-speech tagging, the pos annotator is used to label tokens with their POS tag.

```
String pos = token.get(PartOfSpeechAnnotation.class);
```

Figure 3.8: Sample Code for Getting the Part-of-Speech Tag of Each Token

Part-of-speech tag of each token can be retrieved using the get(PartOfSpeechAnnotation.class) method shown in Figure 3.8. A sample part-of-speech tagging using the online version of Stanford CoreNLP is shown in Figure 3.9.

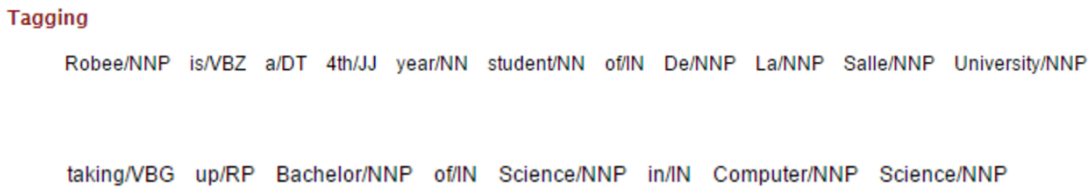

Figure 3.9: Sample Part-of-speech Tagging using Stanford CoreNLP

3.4.3 Parsing

The *parser* annotator generates the parse tree for each sentence. It mainly analyzes the grammatical structure of sentences, for instance, determining which set of words are of the same group and which words are the subject, doer, or receiver of the action.

The sample parser returned by the online version of Stanford CoreNLP from the input is shown in Figure 3.10.

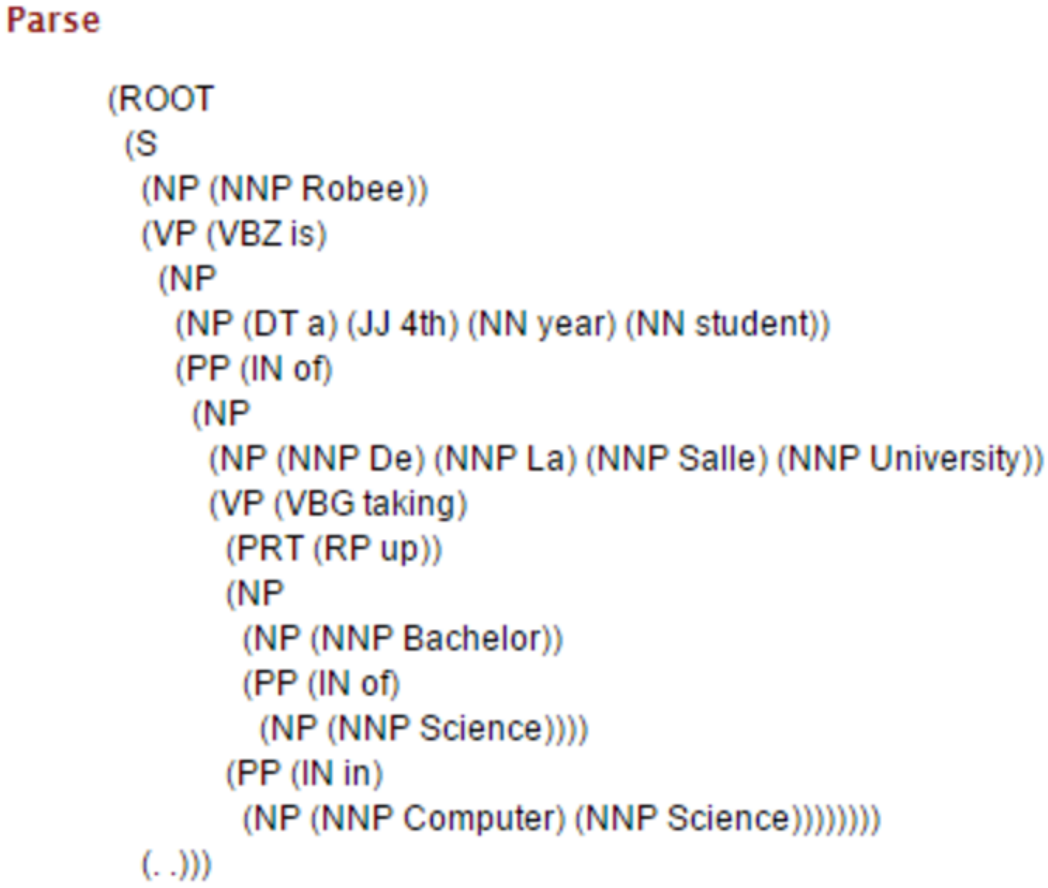

Figure 3.10: Parse Tree Output by Stanford CoreNLP

3.4.4 Universal Dependencies

The Universal Dependencies annotator generates all the possible relationships of one token to the other tokens. It mainly does the full syntactic analysis of the sentence and checks which tokens has a relationship with the others based on the probabilistic parser (Manning et al., 2014).

An example of the universal dependencies returned by the online version of Stanford CoreNLP from the input is shown in Figure 3.11.

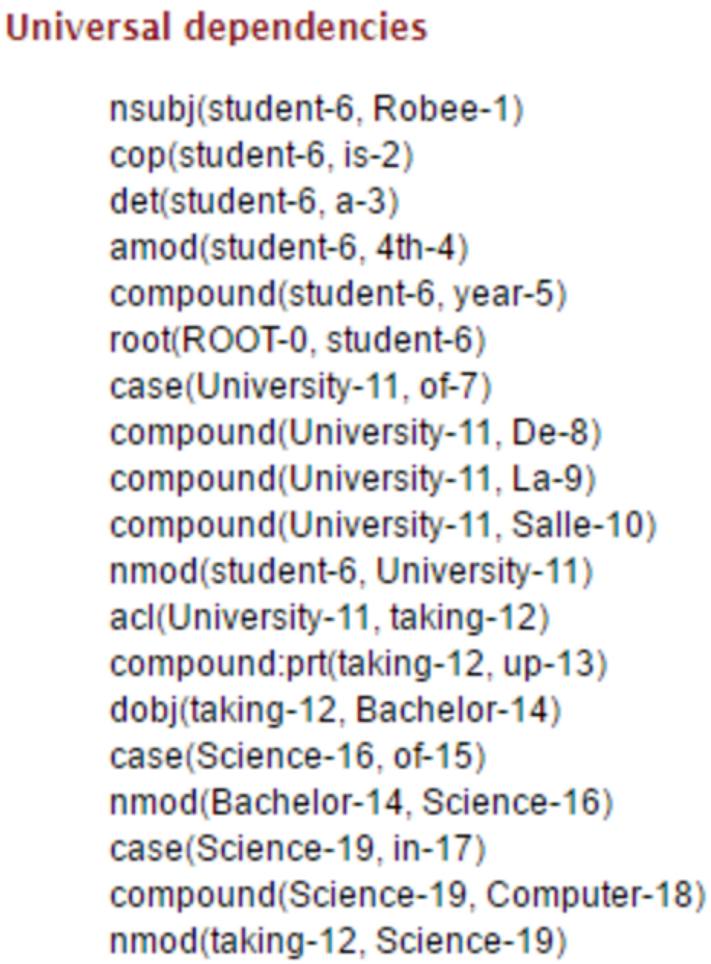

Figure 3.11: Universal Dependencies Output by Stanford CoreNLP

3.5 Text Generation

Natural language generation (NLG) is defined as the process of constructing thoughts or non-linguistic inputs into understandable English texts (Indurkhya & Damerau, 2010; Martin & Jurafsky, 2000; Reiter & Dale, 1997). It uses concepts stored in a knowledge base and decides how to generate text that humans can understand.

An NLG system's task is described as mapping the input data to an output text (Reiter & Dale, 1997). An NLG system must perform six different tasks that needs to be done in order to produce a final output text from the given input. The process in each task is discussed below.

3.5.1 Content Determination

During content determination, the NLG system decides what information needs to be communicated in the text from the given input (Reiter & Dale, 1997). Content must be appropriate for the reader or the user (Martin & Jurafsky, 2000). This process will create messages from the knowledge base. These messages are represented as data objects and will be passed to the next process. The whole message creation process is consists of filtering and summarizing the input data and the messages created can be expressed as entities, concepts and relations in the domain (Reiter & Dale, 1997). In Figure 3.12, three different messages are created.

(1) a.
message-id: msg01
relation: IDENTITY
arguments: arg1: NEXT-TRAIN; arg2: CALEDONIAN-EXPRESS

b. The next train is the Caledonian Express

(2) a.
message-id: msg02
relation: DEPARTURE
arguments: departing-entity: CALEDONIAN-EXPRESS; departure-location: ABERDEEN; departure-time: 1000

b. The Caledonian Express leaves Aberdeen at 10am

(3) a.
message-id: msg03
relation: NUMBER-OF-TRAINS-IN-PERIOD
arguments: source: ABERDEEN; destination: GLASGOW; number: 20; period: DAILY

b. There are 20 trains each day from Aberdeen to Glasgow

Figure 3.12: Sample Messages
(Reiter & Dale, 1997)

3.5.2 Discourse Planning

In comparison to stories having a beginning, middle and end, a message must have a structure, one which must be logical and more distinguishable than the structure used for stories(Reiter & Dale, 1997). In the process of discourse planning, the ordering and structuring of the messages produced by the content determination are done. Content determination has no knowledge of the discourse structure which the message resides nor the content of the message itself (Martin & Jurafsky, 2000).

In generating stories, the system needs to produce a multi-message output. The easiest way is to produce a message for each intended meaning, but what most cases require is the structuring of messages in an appropriate way (Martin & Jurafsky, 2000). For example: "I have just compiled a simple C program. I have just run a single C program. The environment is configured properly." These three messages are individually coherent but are not joined properly.

The discourse planning process results in a tree structure. The leaf nodes represents the individual messages and the internal nodes tells how these messages are placed together and how they are related to one another. Sometimes, the internal nodes also include the discourse relations between their children. For example, as seen in Figure 3.13, the leaf node [DEPARTURE] is an elaboration of the leaf node [IDENTITY]. Later, these clusterings will have an impact in the determination of the sentence and paragraph boundaries of the final output.

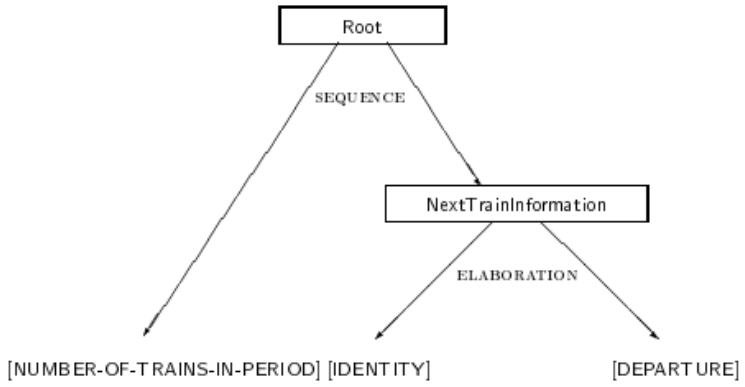

Figure 3.13: Tree Structure Returned by the Discourse Planning (Reiter & Dale, 1997)

The discourse planner is composed of two main components for building the discourse structure namely: (1) Text Schemata and (2) Rhetorical Relations.

3.5.2.1 Text Schemata

A simple method of generating text is positioning words into the templates. Another way to generate texts is to make choices based on the stored data according to the matched patterns in the system's knowledge base (Indurkhya & Damerau, 2010).

An observation of using the text schemata is that the texts follow a structural pattern. The schemata is represented by an augmented transition network (ATN) as shown in Figure 3.14 which consists of states that is represented by an information that is chosen from the collection of data and transitions from one state to the other (Martin & Jurafsky, 2000; Indurkhya & Damerau, 2010). The transition between the states can be described as the cause followed by the effect, a sequence of events, and so on. State S0 defines the start state and State S2 defines the goal state. A loop in the states tells that there are additional information about the object, and sub-steps or side-effects of an action.

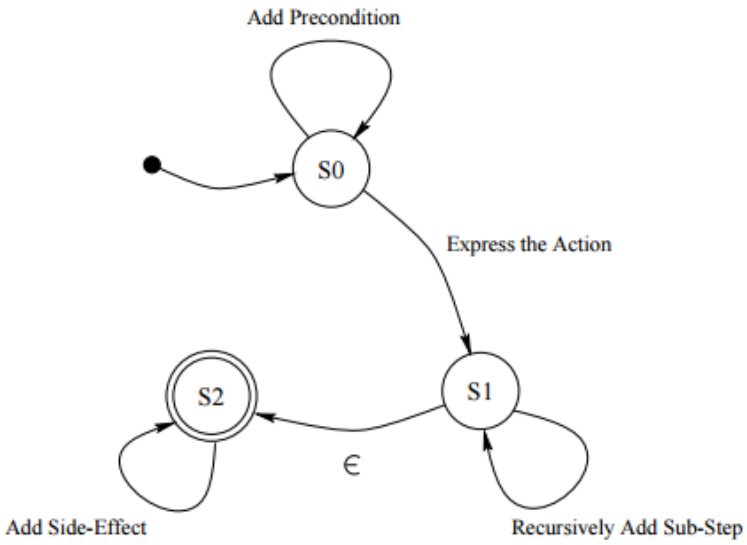

Figure 3.14: Augmented Transition Network (ATN)
(Martin & Jurafsky, 2000)

Using the text schemata is more flexible than directly placing words into the templates. The output was structured according to patterns of expression and add other informations extracted from the knowledge base (Martin & Jurafsky, 2000).

3.5.2.2 Rhetorical Relations

Rhetorical Structure Theory (RST), is a descriptive theory of text organization based on the relationships of two or more messages (Mann and Thompson, 1987 as cited in (Martin & Jurafsky, 2000)). Some of the common RST relations are as follow (Martin & Jurafsky, 2000):

1. ELABORATION - This shows additional details about the content of the message. Elaboration may be in a form of: (1) a member of a set, (2) an instance of an abstract class, (3) a part of a whole, (4) a step of a process, (5) an attribute of an object, and (6) a specific instance of a generalization.
2. CONTRAST - This shows things that, while there are some similarities between the two messages, they are still different in some other ways.
3. CONDITION - This states that something must happen in one of the messages before the situation in the other message can happen.

4. PURPOSE - One of the messages contains the goal of the other message.
5. SEQUENCE - The messages are in arranged in a sequence.
6. RESULT - One of the two messages is an outcome of the other message.

3.5.3 Sentence Aggregation

This process organizes messages together into sentences. Based from the tree output of the Discourse Planning, the leaf nodes [IDENTITY] and [DEPARTURE] are together, so the sentence aggregation can combine the two messages into a single sentence which can be aggregated as "The next train, which leaves at 10am, is the Caledonian Express.". Sentence aggregation is not always required. Some of the messages can be expressed as a single sentence, but it would not result to the readability of the text (Reiter & Dale, 1997).

3.5.4 Lexicalization

In this process, the system maps the right words and phrases to convey the concepts and relations of the messages. The problem in lexicalization is the choosing of appropriate words or phrases to express the content (Martin & Jurafsky, 2000). For example, the words "leave" and' "depart" are both associated with the word "derapture" and the lexical selection must only choose one word to associate departure. Sometimes, lexicalization are simplified by hard-coding a single term with each entry in the knowledge base (Martin & Jurafsky, 2000). Although, this could be improved by varying the words used to express the concept or relation to obtain variety (Reiter & Dale, 1997).

3.5.5 Referring Expression Generation

The task of creating referring expressions to identify the entities are done by the referring expression generation (REG). The Referring Expression (RE) is any noun phrase, whose task is to give identification to the objects (persons, things, or events). REG's goal is to add information to identify the entities unambiguously (Reiter & Dale, 1997). For example, in the sentence: "The next train is the *Caledonian Express. It* leaves at 10am. Many tourist guidebooks highly recommend *this train.*", the entity [CALEDONIAN-EXPRESS] from the content determination is referred to the "Caledonian Express"; the second time to express

Caledonian express is written as the pronoun “it” and the use of “this train” to refer something that was already introduced before. REG and lexicalization are closely associated since both process chooses words and phrases to associated the domain. However, the referring expression generation needs to gather previous information to differentiate one entry from other entries.

3.5.6 Linguistic Realization

Reiter & Dale (1997) explained the last process as applying grammar rules in order to generate a text which is syntactically, morphologically and orthographically correct. The sentence generated is syntactically correct when the grammatical arrangement of words in the sentence is correct; morphologically correct when the structure and form of words are correct; and orthographically correct when the words have correct spelling, hyphenation, capitalization, and punctuation. For example, in the sentence “There are 20 trains each day from Aberdeen to Glasgow”, (1) the syntactic component of the linguistic realizer added the function words “from” and “to” to describe the train’s source and destination, (2) the morphological component of the linguistic realiser converted “train” to its plural form “trains”, and (3) the orthographical component of the linguistic realizer capitalized the first word of the sentence and added a period (.) at the end of the sentence (Reiter & Dale, 1997).

3.5.7 SimpleNLG

SimpleNLG, a Java API for Natural Language Generation, is used to help write a program that can generate grammatically correct English sentences.

```
Lexicon lexicon = Lexicon.getDefaultLexicon();
NLGFactory nlgFactory = new NLGFactory(lexicon);
Realiser realiser = new Realiser(lexicon);
```

Figure 3.15: SimpleNLG’s lexicon, nlgFactory, and realiser

Before the API can generate actual sentences, SimpleNLG lexicon, NLGFactory, and realiser are instantiated (shown in Figure 3.15). Like any other natural language processing systems, SimpleNLG uses information about words from lexicons. SimpleNLG comes with a default lexicon that can be accessed via Lexicon lexicon = Lexicon.getDefaultLexicon(). NLGFactory is used to create SimpleNLG structures, and a realiser is used to transform SimpleNLG structures into text.

```
SPhraseSpec p = nlgFactory.createClause();
```

Figure 3.16: SimpleNLG's SPhraseSpec

To generate sentences, a class called SPhraseSpec is used, which is accessible through the NLGFactory, using the createClause method (shown in Figure 3.16).

```
p.setSubject(pronoun);
p.setVerb(verbObject.getVerb());
p.setObject(verbObject.getNoun());
p.setComplement(verbObject.getTagged());
p.setComplement(verbObject.getLocation());

p.setFeature(Feature.TENSE, Tense.PAST);
```

Figure 3.17: SimpleNLG's SPhraseSpec methods

SPhraseSpec allows for defining a sentence in terms of its syntactic constituents, which can be useful for specifying different parts of a sentence or clause, in no particular order. SimpleNLG then assembles those parts into grammatically correct sentences. The setSubject(subject), setVerb(verb), setObject(object), setComplement(complement), and setFeature(tense, tense) methods are used to define the sentence's subject, verb, object, complement, and tense respectively (shown in Figure 3.17).

```
paragraph = realiser.realiseSentence(p);
```

Figure 3.18: SimpleNLG's realiser

Given the subject - She; verb - going; object - to the mall; tagged - none; location - none; and verb tense - past. The realiser takes in the different components of the sentence, combines them, and generates a syntactically and morphologically correct text using the code snippet shown in Figure 3.18. The resulting sentence will then result to "She went to the mall..

3.6 Knowledge Base

This section discusses the different existing knowledge bases that were used in the implementation of the system: WordNet and ConceptNet.

3.6.1 WordNet

WordNet's database provides connections among nouns and verb synsets containing words that share an underlying meaning. It contains derivational links that connect noun and verb senses (e.g. travel - traveller).

A singleton instance of Dictionary is used to query WordNet using JWNL (shown in Figure 3.19).

```
JWNL.initialize(new FileInputStream("src/main/resources/properties.xml"));
final Dictionary dictionary = Dictionary.getInstance();
```

Figure 3.19: Instantiation of the Dictionary

Afterwards, a lemma can easily be queried from the dictionary (e.g. travel). For each lemma, there can be one part-of-speech specified among four possible part-of-speech classes: *POS.ADJECTIVE*, *POS.ADVERB*, *POS.NOUN*, and *POS.VERB*.

The system checks if the lemma is in the dictionary. If the lookup fails, *indexWord* is null as shown in Figure 3.20.

```
IndexWord indexWord = dictionary.lookupIndexWord(POS.VERB,"travel");
```

Figure 3.20: Setting the lemma and part of speech

Different senses that the lemma may have are retrieved as shown in Figure Figure 3.21.

```
final Synset[] senses = indexWord.getSenses();
```

Figure 3.21: Getting related senses

For each sense, a short description of the sense called *gloss* is derived as shown in Figure 3.22.

```
final String gloss = synset.getGloss();
```

Figure 3.22: Getting short description of senses

Other more specific lemmas under each sense are also derived as shown in Figure 3.23.

```
final Word[] words = synset.getWords();
```

Figure 3.23: Getting words under specific senses

WordNet returns senses containing related lemmas as shown in Figure 3.24.

```
For sense: 1 (change location; move, travel, or proceed, also metaphorically; "How fast does your new car go?";
        travel([POS: verb])(travel%2:38:00::)
        go([POS: verb])(go%2:38:00::)
        move([POS: verb])(move%2:38:03::)
        locomote([POS: verb])(locomote%2:38:00::)
For sense: 2 (undertake a journey or trip)
        travel([POS: verb])(travel%2:38:04::)
        journey([POS: verb])(journey%2:38:00::)
For sense: 3 (make a trip for pleasure)
        travel([POS: verb])(travel%2:38:01::)
        trip([POS: verb])(trip%2:38:02::)
        jaunt([POS: verb])(jaunt%2:38:00::)
For sense: 4 (travel upon or across; "travel the oceans")
        travel([POS: verb])(travel%2:38:02::)
        journey([POS: verb])(journey%2:38:01::)
For sense: 5 (undergo transportation as in a vehicle; "We travelled North on Rte. 508")
        travel([POS: verb])(travel%2:38:03::)
For sense: 6 (travel from place to place, as for the purpose of finding work, preaching, or acting as a judge)
        travel([POS: verb])(travel%2:38:05::)
        move_around([POS: verb])(move_around%2:38:00::)
```

Figure 3.24: Related senses returned by WordNet

By using the mapping available on WordNet's database, morphosemantically-related words are extracted as shown in Figure 3.25.

```
For sense: 1 (change location; move, travel, or proceed, also metaphorically; "How fast does your new car go?";
        mover%1:18:00::; someone who moves
        traveller%1:18:00::; a person who changes location
        traveler%1:18:00::; a person who changes location
        locomotion%1:04:00::; self-propelled movement
        motion%1:04:01::; the act of changing location from one pl
        locomotion%1:07:00::; the power or ability to move
        movement%1:04:04::; the act of changing location from one pl
        movement%1:11:00::; a natural event that involves a change i
        move%1:04:04::; the act of changing location from one pl
        travel%1:11:00::; a movement through space that changes th
        travel%1:04:01::; self-propelled movement
For sense: 2 (undertake a journey or trip)
        journeyer%1:18:00::; a traveler going on a trip
        traveller%1:18:00::; a person who changes location
        traveler%1:18:00::; a person who changes location
        journey%1:04:00::; the act of traveling from one place to a
        travel%1:04:00::; the act of going from one place to anoth
For sense: 3 (make a trip for pleasure)
        traveller%1:18:00::; a person who changes location
        traveler%1:18:00::; a person who changes location
        tripper%1:18:00::; a tourist who is visiting sights of inte
        jaunt%1:04:00::; a journey taken for pleasure; "many summ
        trip%1:04:00::; a journey for some purpose (usually incl
For sense: 4 (travel upon or across; "travel the oceans")
        journeyer%1:18:00::; a traveler going on a trip
        traveller%1:18:00::; a person who changes location
        journey%1:04:00::; the act of traveling from one place to a
        travel%1:04:00::; the act of going from one place to anoth
For sense: 5 (undergo transportation as in a vehicle; "We travelled North on Rte. 508")
        traveller%1:18:00::; a person who changes location
        traveler%1:18:00::; a person who changes location
        travel%1:11:00::; a movement through space that changes th
For sense: 6 (travel from place to place, as for the purpose of finding work, preaching, or acting as a judge)
        traveller%1:18:00::; a person who changes location
        traveler%1:18:00::; a person who changes location
        travel%1:04:00::; the act of going from one place to anoth
```

Figure 3.25: Morphosemantically related words returned by WordNet

3.6.2 ConceptNet

ConceptNet is a semantic network containing nodes or concepts which are represented by words or short phrases (e.g. "ball", "toy") and the semantic relations between them (e.g. "IsA", "PropertyOf") [4]. This contains things that computers should know about the world specifically when understanding text written by human. The relationships labeled between them will help computers in searching information, answering questions and understanding humans. ConceptNet contains everyday basic knowledge, cultural knowledge, and scientific knowledge. ConceptNet also supports language like Chinese and Japanese.

Previous data from ConceptNet are from a home-grown crowd-sourced project which a website is ran to collect facts from humans who visits the site. The ConceptNet 5.0's knowledge base currently contains 12.5 million edges, representing 8.7 million assertions and connecting 3.9 million concepts into a semantic network of more than 2.78 million nodes which are classified into 23 semantic relations as discussed in ConceptNet5 wiki website (Speer & Havasi, 2012).

ConceptNet5 [5] has a REST API which according to ConceptNet allows the user to: (1) retrieve the data from the nodes and edges, (2) query the edges given a property, and (3) measure and query the semantic distance between two nodes.

3.7 Evaluation Metrics

Evaluation metrics are used to evaluate the effectiveness of computer systems and to justify its developments of these systems (Pehcevski & Piwowarski, 2009). They are necessary to have a formal method of objectively determining the shortcomings of the system in order to be able to improve it. For this research, the three general topics to tackle in the evaluation metrics are [1] the content and quality of the story; [2] the style and appropriateness of writing; and [3] the algorithms under the hood that worked to provide the generated story.

In this research, the primary component to be evaluated is the quality of the resulting life stories, in terms of completeness. When evaluating the writing of a beginner such as a human child, an article by (Hamilton, n.d.) suggests that the focus should be on the content of the story rather than mechanical errors such as those involving spelling or punctuation. As mentioned in 3.1.1 Elements and Structure of a Story, some elements of a life story may not be present in this

[4] http://conceptnet5.media.mit.edu/
[5] http://conceptnet5.media.mit.edu/

generated story due to the age of the person being talked about. However, there is still plenty of content available in Facebook data.

The life story should be composed of three main parts: [1] the introductory part, which introduces the person and presents some interesting facts about him/her; [2] the body part tells stories about the person that had happened in his/her life; and finally, a [3] conclusion part that describes the person's preferences and likes. A life story is deemed complete if these parts are present in the story and can be easily identified by the reader.

The introductory part should contain all the basic information about the user such as the name, birthdate, birthplace, and other profile information that are available. It should correctly follow the structure of the templates described in Appendix J, Knowledge Representation, and filled with the correct data. The body part should contain relevant posts relating to one's life events. This requires validating the algorithms used for post selection and story generation (specifically content determination and sentence aggregation). Finally, the conclusion should specify five of the user's top preferences, with some other sentences describing how much this person likes those preferences.

In addition, the events specified in the generated story should be traceable from the data extracted from the user. The user should be able to infer that an event in the story came from a specific post that indicates that this event happened. This can be made easier by including time elements in the generated story (e.g. "last year", "two months ago"). The correctness of the content determination would be dependent on the quality and amount of data that can be extracted from the user's Facebook posts.

Another consideration for the evaluation metrics is analyzing the style of writing, as inspired by the works of (Staff, 2000). Do the words flow together nicely, for example? For this question, the story should be checked to determine if lexical choices consistently use the correct words, including the discourse markers used to connect the different events and posts to generate coherent stories. Also, is there a sense of organization and focus in the writing of the story? Does it have a strong beginning and a good ending? Are there enough details provided to describe the person? Finally, is the writing free of misspellings, wrong capitalizations, wrong tense use, and wrong use of punctuations?

Lastly, another consideration of the evaluation metrics should be the system architecture itself – the different algorithms used by the software (described in Chapter 4, particularly in 4.4, Architectural Design). Every process must be validated, in terms of functionality and quality. For example, did the system extract all needed data and store them correctly? Did the text understanding

API return the correct entities (as described in Section 3.4.1, Entity Recognition) and did so correctly? Is the knowledge base present and working properly, and was it shown in the generated story that it worked properly and was utilized correctly? Finally, did the text generation process generate a story that meets the primary criteria of the evaluation metrics– content and style?

Results from evaluation can then be used to improve the quality of the output of this research, which will be described in the next chapter: the system design.

Chapter 4

FB Stories

This chapter discusses the functional requirements and the overall specifications of the software developed as part of this research, called FB Stories . It introduces the software, its objectives, scope and limitations, architecture, and features.

4.1 An Overview

FB Stories is a web-based application where Facebook users can generate their own life stories through the data gathered from their Facebook account, with the use of natural language processing (NLP) and generation (NLG) techniques.

To use FB Stories , the user must log in to their Facebook account to allow access to their data. The data extracted will be classified as either direct knowledge or indirect knowledge. For indirect knowledge, text understanding techniques will be applied.

The software then proceeds to the text generation module, where it determines which elements of the extracted data are appropriate and can be used in generating the life story, and constructs corresponding story text from these data. Once the life story text has been generated, the user is given the option to save the story into a text file.

The completed life story contains three parts. The first part contains basic information or facts about the user. The second part contains data extracted from the Facebook posts that were classified and stored into the indirect knowledge base. The third part contains the list of preferences of the user inferred from

the available list of page likes, as well as some recent events that this person has attended.

4.2 Software Objectives

This section presents the general and specific objectives of FB Stories .

4.2.1 General Objective

To generate a story that takes into account the Facebook posts of a user by using natural language processing techniques.

4.2.2 Specific Objectives

1. To extract needed data from Facebook;
2. To use data processing techniques to analyze the input;
3. To classify each post according to its type;
4. To use text generation techniques to generate a story;
5. To allow users to save the generated stories into a text file.

4.3 Scope and Limitations of the Software

4.3.1 Data Extraction

In retrieving data from a user's Facebook account, asking for user's permission is necessary. A successful login to the user's Facebook account means that the user permits the system to make use of their data. These permissions will be readily set out for the user to approved, and once set, selected options cannot be altered. If the user does not allow the software to access his/her profile with the given permissions, only those public information and public posts of the user are extracted.

FB Stories does not extract data from Messenger, as well as information about the user's interactions (such as who Liked the user's posts).

4.3.2 Data Processing

Data processing techniques are only applicable to Facebook posts. The system does not perform any verification on the correctness of the user's data.

Hashtags, hyperlinks, emoticons, laughter, and foreign characters are removed from posts before undergoing text understanding, in order to lessen misclassifications. For posts with mixed languages, abbreviations and incomplete sentences, the interpretation of these are limited by what the API can offer Figure 4.1. The syntax analysis returned by the tool is not checked for the correctness of its output. In Figure 4.1, "kain is a Filipino word which means "eat, thus it should be a verb not a noun.

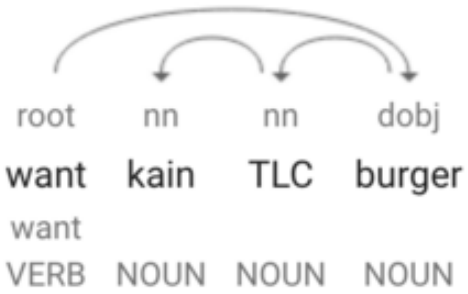

Figure 4.1: Sample Incomplete Text with Mixed Languages and Abbreviation.

4.3.3 Post Classification

Posts extracted from Facebook cannot be used directly in generating story text, since users tend to post snippets of incomplete, context-based data. Other posts have no explicit verbs used to describe the event. Thus, the posts must be individually processed to extract the necessary information comprising an event.

Although Facebook has a feature called *Predefined Activities* to enable users to easily classify their individual posts according to the content, there are currently no tools that can support the extraction of relevant elements from posts that uses this feature. Thus, a post classification algorithm is needed to classify each Facebook post as either *celebrating* post, *travelling* post, *drinking* post, *eating* post or *no event* post. These four types of posts are chosen to be used in this

research since majority of Facebook posts gathered and analyzed fall under them. Other types of posts such as *reading*, *listening*, *watching*, among others may be present in the extracted Facebook posts, but will be ignored and will be tagged as *no event* posts for this research.

4.3.4 Text Generation

Text generation has three components. One component is responsible for generating the introductory part of the story text, the second component is responsible for generating the body and the last component is responsible for generating the conclusion.

The introduction contains basic information directly extracted from the user's profile without going through further processing. The body part of the life story contains events: information which underwent processing and post classification. The conclusion part of the life story contains the likes and preferences of the user, as well as recent Facebook Events attended.

4.3.5 Save to Text File

The user might want to cherish and read previously generated life stories in the future, making it important to have a method to save stories. Text files are only generated based on the user's instructions. Other file types than *.txt* are not supported. After successfully saving the generated story into a text file, the user is solely responsible for the safekeeping and/or dissemination of the file.

For purposes of validating the output of FB Stories and for future research, a copy of the generated life story is also stored as part of the output of this research. The user is properly informed of this. Anonymizing the data for future use is provided as an option for confidentiality

4.4 Architectural Design

Figure 4.2 is a representation of the architecture design of FB Stories . It is divided into three big parts: initialization, text understanding, and text generation.

A more detailed discussion of the different modules is written in Chapter 5 for both the initial version of the system as well as the latest version as of the time of writing.

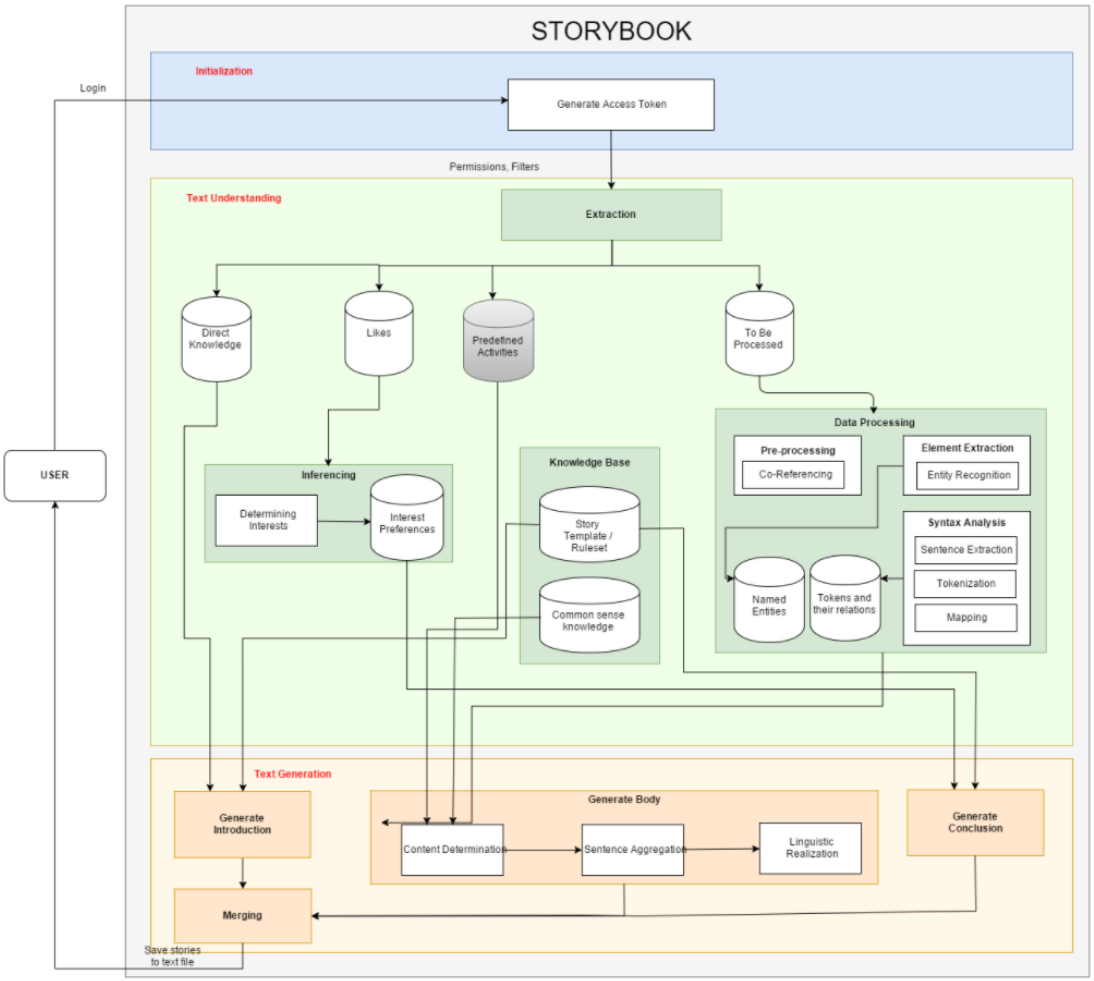

Figure 4.2: Architecture Design of FB Stories

4.4.1 Initialization

Before the system can extract the user's data, it needs the user's permission, which the user grants by logging into Facebook and specifying that they agree with the data extraction to be done by FB Stories. After the user goes through each of the permissions and accepts it, Facebook generates an access token, which controls which data can be extracted by the data extraction tool.

The data extracted can be classified into one of three types:

- *Direct knowledge.* These are derived from the user's *About Me* section, and are used for generating the introductory part of the life story.
- *Data to be processed.* These are taken from the user's posts on their Timeline, and will be processed later on for use in generating the body.
- *User's list of Liked pages and Facebook Events.* These are derived from the user's account (Facebook keeps track of the user's list of Liked pages and Facebook Events attended). These will be used as part of the conclusion.

4.4.2 Text Understanding

For those data from which knowledge cannot be easily inferred, a more specific procedure of text understanding algorithms is applied. Data preprocessing processes the input in order to deal with issues present in user-generated data, such as stray characters or the presence of laughter.

After preprocessing these data, they are then subjected to NLP processes which enable the system to figure out the relevant parts of these data to use in generating the story later on, such as who did what, what is being done, and to whom or to what. These data are then classified (with the help of the knowledge base) in order to figure out how they'll be organized in the final generated story.

4.4.3 Text Generation

The text generation module is responsible for generating the appropriate story segments that make up the entire life story generated by FB Stories. There are three components: the *GenIntro*, *GenBody*, and *GenConclusion*.

GenIntro uses data determined to be "facts or direct knowledge. It involves checking the knowledge base for the appropriate template to be used based on the available facts in the *Direct Knowledge*, *Educational Background*, *Work*, and *Family* table. GenIntro would also involve filling the template with the correct data. The generated text will become the introductory paragraph of the life story.

GenBody, on the other hand, is applied to generate the body of the life story from the data processed in the Data Processing module. The body paragraph(s) will be narrating one or more events about a person's life. These events will be taken from data that users have written and posted on their own Timeline. For these data, text generation is more complex. It will undergo three sub-modules, which are all detailed in Section 3.5.1, Text Generation.

GenConclusion will be used to generate the conclusion part of the story text, which will contain the user's likes and interested events. The exact process of determining the user's likes is explained in the Inferencing Section, 4.4.4. Similar to the approach in GenIntro, GenConclusion would check the available templates defined in the Template table and use the data stored in the *Likes* and *Events* table.

The generated texts from the three text generation modules are then merged, and presented to the user as the complete life story. The user will now have the option to save or discard their life story. If the user wishes to save their life stories, a text file will be created and the user chooses where to save this file. But if the user wishes to discard this, then they simply close the software.

4.5 Software Functions

FB Stories provides a simple environment that allows users to easily use it to create a life story from their Facebook posts and save these stories for future use. Below are the software functions used in FB Stories .

4.5.1 Login Window

In using the application, the user would have to Login to Facebook for the software to access the data stored in his/her account. A Facebook login button would prompt the user to login as shown in Figure 4.3. Once the button is clicked, a Login Window, which can be seen in Figure 4.4, would pop up informing the user that he/she is logging in his/her Facebook account with the app, StoryBook.

Login information such as email address, phone number or user id along with the corresponding password are both needed to successfully login.

Logging in to the app also grants the permissions set by the app which then automatically updates the access token stored in cookies.

Figure 4.3: Facebook Login Button.

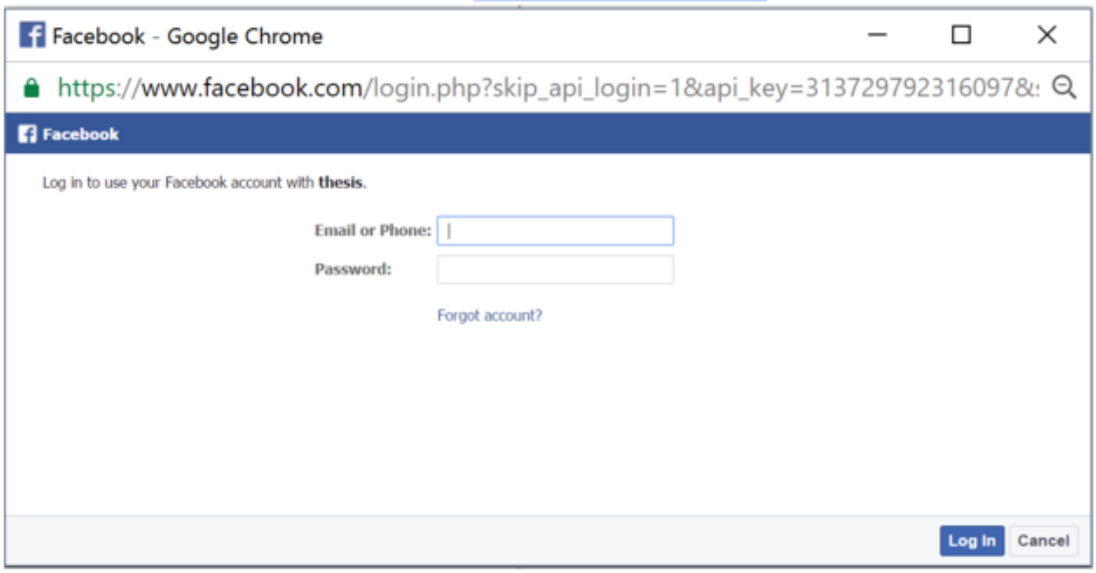

Figure 4.4: Pop-up Login Window.

4.5.2 Permission Window

Logging in to Facebook automatically stores a default access token in cookies. If the user is already logged-in on Facebook and the FB Stories automatically obtains the user's access token, then it would check permissions in the access token and prompts the user of the missing permissions it needs to acquire. Figure 4.5 displays the permission window that asks the user to grant the permissions needed by the software.

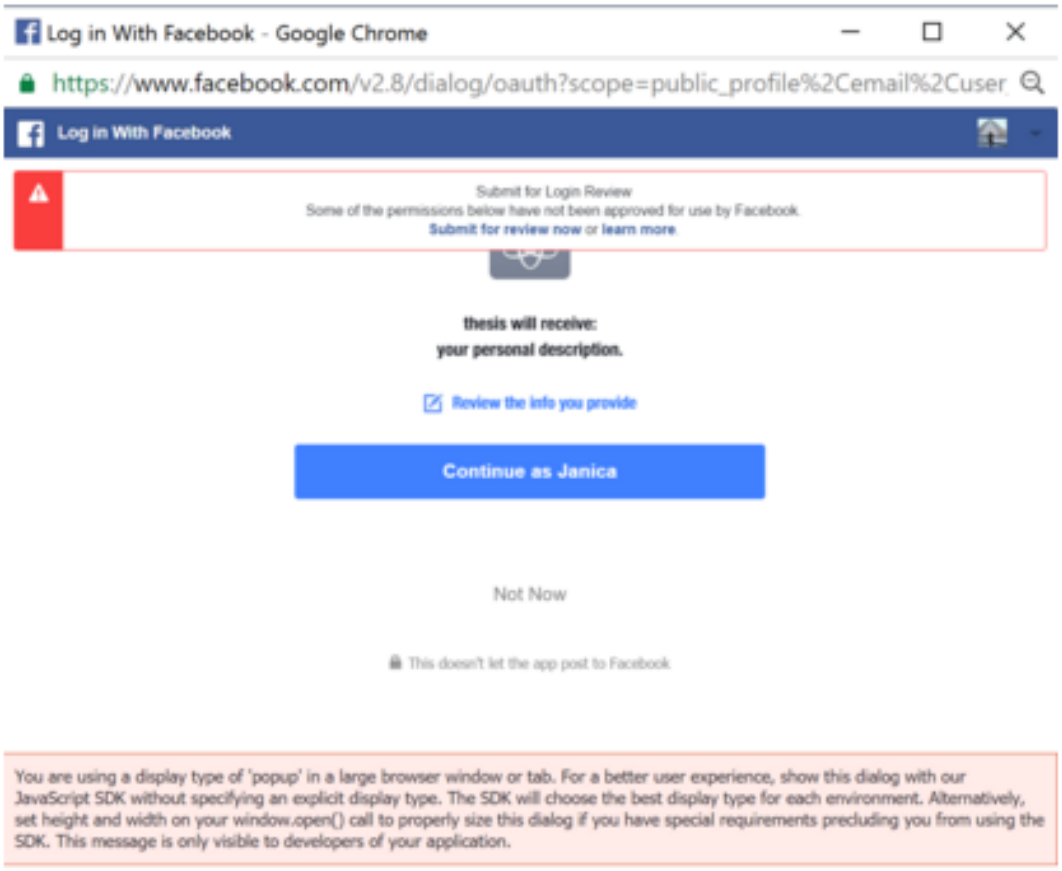

Figure 4.5: Permission Window.

4.5.3 Generated Story Output Window

The Generated Story Output Window, Figure 4.6, is basically a window that displays the generated story after FB Stories has analyzed and processed the gathered data from the user's Facebook account. The user would be given an option to save the generated story to a text file through the "Save to Text File" button.

Figure 4.6: Generated Story Output Window.

4.6 Physical Enviroment and Resources

This section details the minimum and recommended software requirements for software implementation.

FB Stories requires the following resources for development:

- Eclipse IDE
- MySQL Server and Workbench
- Java Development Kit (JDK)
- Apache Tomcat 8.0

These are the minimum software requirements for FB Stories to run properly:

- **OS**: Windows 8.1 / 10
- **Memory**: 1 GB RAM
- **Storage**: At most 1 GB of free hard disk space
- **Internet connection**: Broadband, at least 1Mb/s bandwidth
- **Others**: Java Runtime Environment (JRE), MySQL Server, Apache Tomcat 8.0

4.6.1 Tools

The following tools will be used for the development and runtime of the FB Stories .

4.6.1.1 Facebook Login API

The Facebook Login API enables the use of the Facebook user's identity in order to craft interesting stories about them. It enables the application to extract data from Facebook to be processed and used to generate stories. Features of Facebook Login, such as access tokens and permissions, make it safe and secure for people and apps to use, but there are some security steps which this software will need to implement. This will be tackled in Chapter 5, Design and Implementation.

4.6.1.2 Graph API

The use of Graph API enables the software to extract posts and data from a specific Facebook account. It supports developers by supplying services such as providing snippets of codes for easier integration with JSON requests and responses.

4.6.1.3 Stanford CoreNLP

Stanford CoreNLP will be used in the text understanding module, as it provides the needed component, syntax analysis.

4.6.1.4 WordNet

WordNet will be used to supply the data that is needed for the reference table which contains keywords such as related verbs and nouns. The reference table will then be used in the classification of Facebook post which does not contain a verb.

4.6.1.5 ConceptNet

ConceptNet is a semantic network containing concepts with Open Mind Common Sense as its main source of knowledge, along with other sources such as Wikipedia, WordNet, and DBPedia. This knowledge will be used to supply keywords for the reference table to be used in the post classification module. This will gather related verbs and noun for the categories *Celebrating*, *Travelling*, *Eating*, and *Drinking*.

4.6.1.6 SimpleNLG

SimpleNLG will be used in generating grammatically correct English sentences. SimpleNLG will also automate some of the tasks an NLG system needs to perform like checking the orthography, morphology and simple grammar of the sentences.

Chapter 5

Design and Implementation

This chapter presents the design and implementation of StoryBook. It first discusses the authors' journey in dealing with the data in Facebook and the issues inherent in its characteristics. This is followed by the discussion on the event classification algorithm of the system, its issues, initial shortcomings, and improvements. It ends with a discussion of the natural language generation module, which has three submodules: the GenIntro, GenBody, and GenConclusion. The issues encountered during implementation and the solutions applied to address each issue are presented.

5.1 System Design

Figure 5.1 shows the system architecture of FB Stories .

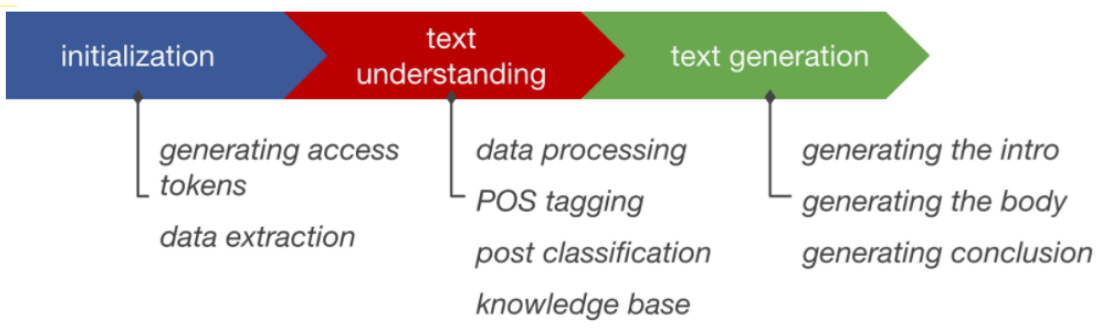

Figure 5.1: System Architecture of FB Stories

5.1.1 Initialization

Part of: Startup

The user, via the use of the Facebook Login API (Facebook, n.d.-b), must give his/her login credentials to allow their Facebook data to be extracted for use. After the user goes through each of the permissions and allows them, Facebook generates an access token, which is then used by Graph API to determine which data can be extracted from the Facebook account of the user.

5.1.2 Data Extraction

Part of: Text Understanding

This uses Graph API. Three types of data need to be extracted and partitioned into: [1] personal info that can be used as is, such as the user's birthday and list of family members; [2] data which have to be processed such as posts; and [3] the user's list of preferences and events attended.

The partitioned data will be stored in seven separate tables: direct_knowledge, educational_bg, work, family, to_be_processed, likes and events, as shown in Figure 5.2.

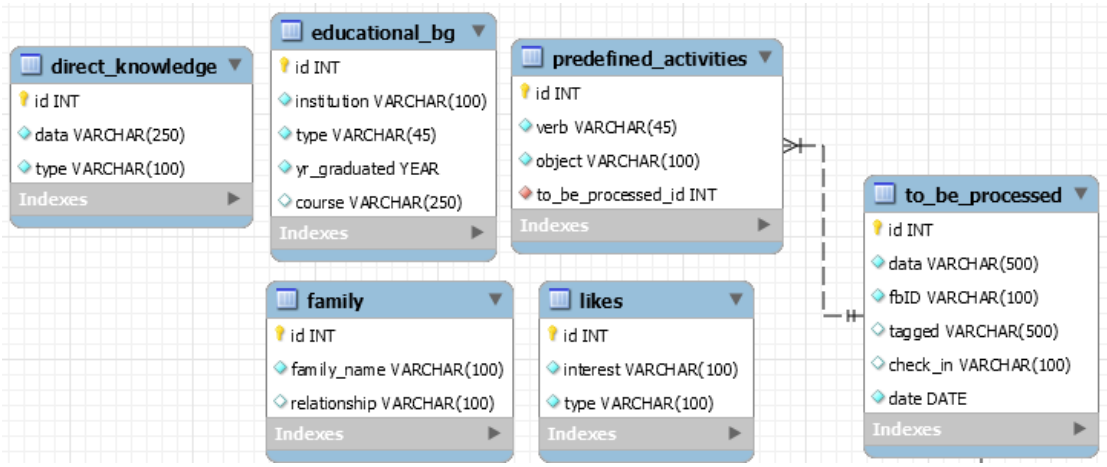

Figure 5.2: Database Design for Storing Extracted Data.

Direct knowledge can be extracted from Facebook's About Me section, which contains the personal information of the user. The extracted data also known as

the direct knowledge or facts about the user are stored in the *Direct Knowledge Table* (Table 5.1), which contains the following fields:

- id - unique value that identifies each fact
- data - the direct knowledge data extracted from his/her Facebook account
- type - the type of knowledge that describes the data

See Appendix J for the complete list of types available and a description of each.

Given the sample *About Me* section of a user in Facebook (Figure 5.3), some of the data are shown in Table 5.1:

Figure 5.3: Sample About Me Section in Facebook

Table 5.1: Sample data in the direct_knowledge table.

id	data	type
1	Robee Khyra Te	name
2	NULL	middle_name
3	Te	last_name
4	1996-05-25	birth_date
5	Manila, Philippines	location

The educational background of the user can also be extracted from Facebook's *About Me* section. These are then stored in the *Educational Background Table* (Table 5.2), which contains the following fields:

- id - unique value that identifies each educational level
- institution - the name of the institution
- type - the type of educational level
- year_graduated - year graduated or ended in the institution

- course - course taken in the said institution
- fbID - unique id generated by Facebook for each institution

See Appendix J for the complete list of types available and description of each.

Table 5.2: Sample data in the educational$_b$*gtable*.

id	**institution**	**type**	**year_ graduated**	**course**	**fbID**
1	De La Salle University	College	null	Bachelor of Science in Computer Science with specialization in Software Technology	
2	Chiang Kai Shek College	High School	2013	null	

Current or previous work of the user can also be extracted from Facebook's About Me section. These are then stored in the *Work Table* (Table 5.3), which contains the following fields:

- id - unique value that identifies each work
- institution - the name of the institution
- date_started - year started working in the institution
- date_ended - year ended working in the institution
- location - location of the said institution
- fbID - unique id generated by Facebook for each work institution

Family members of the user can also be extracted from Facebook's *About Me* section. These are then stored in the *Family Table* (Table 5.4), which contains the following fields:

- id - unique value that identifies each family member
- family_name - name of the family member

- relationship - relationship with the user
- fbID - unique id generated by Facebook for each family member

See Appendix J for the complete list of available relationships.

Data taken from the user's posts, on the other hand, requires further processing.

The assumptions for these posts are:

a. They all contain a text portion;

b. The created time of the original post (in case there were no edits made) is assumed to be the time the event happened;

c. If there were edits made, the edited time would be used and considered to determine the time sequence of the posts;

d. The people tagged are used to know that those people are with the user at the time of the event; and

e. All content provided are correct.

Table 5.3: Sample data in the work table.

id	**institution**	**date_ started**	**date_ ended**	**location**	**fbID**
1	University Student Government, DLSU	2013-09-01	2014-04-30	Manila, Philippines	3459628907221

Table 5.4: Sample data in the family table.

id fbID	**family_name**	**relationship**
1	Jennilyn Wang	sister
2	Renee Te	sister

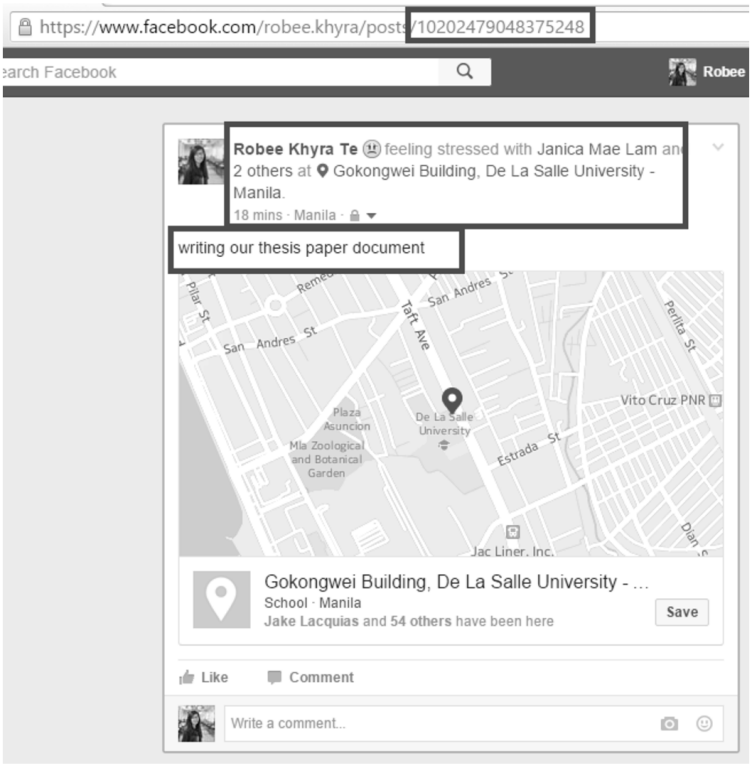

Figure 5.4: Sample Post in Facebook

The *To Be Processed Table* (Table 5.5) is used to store the description or caption in the user's status and posts, including other relevant information regarding the posts. The following are the attributes of the *To Be Processed* table:

- id - unique value that identifies each post
- data - the description/caption extracted from his/her Facebook account
- fbID - unique id generated by Facebook for each post
- tagged - comma-separated values representing the friends the user is with for each post
- place - the place where the event happened
- city - the city where the event happened
- country - the country where the event happened
- year - the year when the post was created
- month - the month when the post was created

- day - the day when the post was created

Table 5.5: Sample data in the to_be_processed table.

id	data	fbID	tagged	check_in	date
1	writing our thesis paper document	102024790 48375248	Janica Mae Lam, Camille Saavedra, Alds Hade	Gokongwei Building, De La Salle University - Manila	2016-11-22

Given the sample list of user's liked pages section in Facebook (Figure 5.5), the liked data that are stored in the database is shown in Table 5.6.

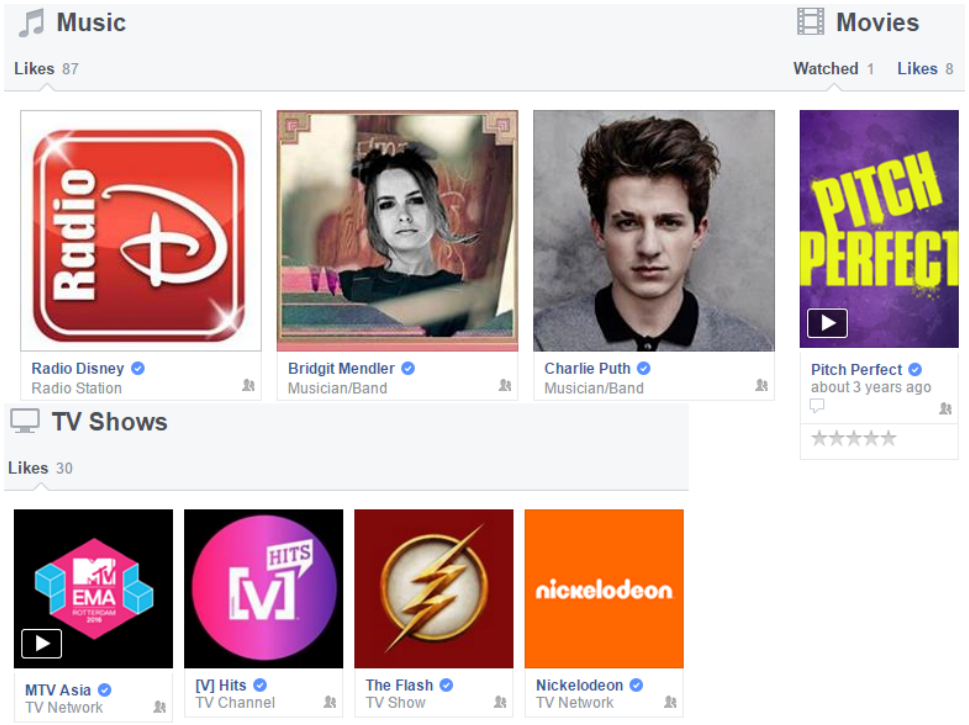

Figure 5.5: Sample Interest Preference in Facebook

The *Likes Table* (Table 5.6) contains the interest preferences of a particular user in Facebook. This also specifies what time of interest it is.

- id - unique value that identifies each interest
- interest - the specific interest liked by the user in his/her Facebook account

- type - the type of interest that describes the interest
- fbID - unique id generated by Facebook for ech page

See Appendix J for the complete list of available types.

Table 5.6: Sample data in the likes table.

id	**data**	**type**
1	Radio Disney	121 (Radio Station)
2	Bridgit Mendler	93 (Musician/Band)
3	Charlie Puth	93 (Musician/Band)
4	Pitch Perfect	91 (Movie)
5	MTV Asia	147 (TV Network)
6	[V] Hits	146 (TV Channel)
7	The Flash	149 (TV Show)
8	Nickelodeon	147 (TV Network)

Given the sample list of user's going and interested events in Facebook Figure **??**, the events data that are stored in the database is shown in Table **??**.

The *Events* table (Table **??**) contains all the going and interested events of a particular user in Facebook. It contains the following fields:

- id - unique value that identifies each interest
- name - the specific interest liked by the user in his/her Facebook account
- rsvp_status - the type of interest that describes the interest
- place - the place where the event took place
- city- the city where the event took place
- country - the country where the event took place
- fbID - unique id generated by Facebook for each event

See Appendix I for the complete list of types available.

5.1.3 Inferencing

Part of: Text Understanding

Some user interests have to be inferred, to answer questions such as, "how does one determine if one likes music? This particular question can be answered by looking at the list of a person's liked pages.

Facebook has six (6) general categories for Pages, and around 100 specific categories. For this study, the top five categories with the most liked pages by the user are written down in the story. Narrowing down the categories to the top five allows the life story to focus on the things that the user likes the most.

For instance, if the user likes 87 pages of the category "Musician/Band and it falls under the top five categories of pages she has liked, it can be inferred that she likes music. The top five categories are later used in the conclusion. 2-3 sample pages under each category are used in the to provide support.

5.1.4 Data Processing

Part of: Text Understanding

This process utilizes Stanford CoreNLP. For those data from which knowledge cannot be easily inferred, a more specific procedure of text understanding algorithms is applied. Data preprocessing processes the input and stores them in an abstract representation for use by other modules of FB Stories . Hashtags, hyperlinks, emoticons, laughter, and foreign characters are removed from posts before undergoing text understanding, in order to lessen misclassifications.

After the text has been preprocessed accordingly, different event details such as the noun phrase and verb phrase can now be identified. Stanford CoreNLP splits a post into sentences, and for each sentence, syntax analysis is performed.

The direct objects, the lemmatized verb, as well as other information that were extracted directly from Facebook earlier such as the date of the post, location of the event, and people whom the user is with at the time of the post are then stored in the *Verb Object Table* (shown in Table 5.7) to be used later for the generation of body.

The *Verb Object* table (Table 5.7) contains all the event details in a particular post. It contains the following fields:

- id - unique value that identifies each interest
- post_id - value (corresponding to the id from the to_be_processed table) representing the post where the sentence is taken from
- verb - the identified verb of the post
- noun - the identified object of the post
- sentence - an individual sentence from the original post
- post_type - value (corresponding to the id from the post_type table) representing the post type of the sentence
- tagged - the people tagged in the post acquired from the to_be_processed table
- location - the location where the event took place acquired from the to_be_processed table
- date - the date when the event happened acquired from the to_be_processed table

Table 5.7: Sample data in the verb object table.

id	post_id	verb	noun	sentence
1	1	write	thesis paper document	writing our thesis paper document

post_type	tagged	location	date
	Janica Mae Lam, Camille Saavedra, Alds Hade	Gokongwei Building, De La Salle University	11/22/2016

Notice that the post_type is currently empty because the post still needs to undergo the post classification module. In cases that the Stanford CoreNLP cannot determine a verb or a noun from the given post, the column verb or noun will also be empty. Also, the columns tagged and location can also be empty, if there are no data supplied by the user.

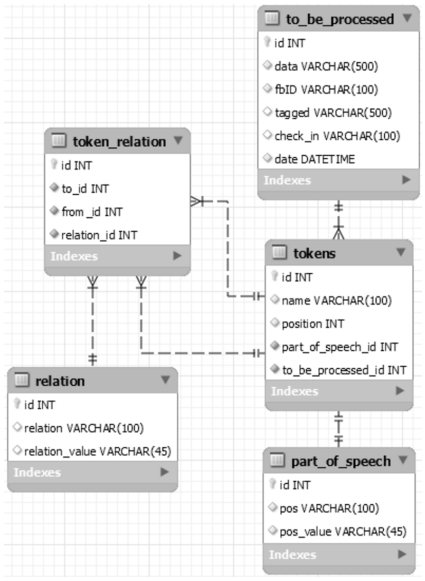

Figure 5.6: Database design for storing the processed data.

5.1.5 Knowledge Base

GenIntro, *GenBody*, and *GenConclusion*, which will be discussed in 5.1.6, uses grammar rules and assertions in order to generate the story. These are stored in the database.

5.1.6 Text Generation

The text generation module is responsible for generating the appropriate story segments that make up the entire life story generated by FB Stories . The tool, SimpleNLG, is used for text generation.

There are three components of the text generation modules, namely the GenIntro, GenBody, and GenConclusion. GenIntro is applied to the data determined to be "facts" or direct knowledge. The generated text will become the introductory paragraph of the life story.

GenBody is applied to generate the body of the life story from the data processed in the Data Processing module. The body paragraph(s) will be narrating one or more events about a person's life. These events will be taken from data that users have written and posted on their own Timeline. For these data, text generation is more complex. It will undergo three sub-modules, which are all detailed in Section 3.5.1, Text Generation. Each activity in each sub-module is discussed below.

From the data forwarded by the Pre-Processing module, content determination will now classify all the verbs stored in the *Verb Object* table according to type (e.g. all "eating together, then all "travelling to, then all "celebrating). Because a single post can contain multiple verbs or words signifying events, it can be classified into multiple categories. For example, the post, "*walking around the streets of Rome while eating delicious gelato.* is classified as eating and travelling.

From the given sample post above, the verb is identified to be "write. To be able to determine the post type, it will check the *Post Type* table for the corresponding id of the verb and update the post_type column in the *Verb Object* table. After the post classification module, the *Verb Object* table (shown in Table **??**) now contains the updates data.

However, in case of there is no verb present in the post, a reference table (shown in Appendix M) containing predefined keywords commonly associated with each event category that was derived through manual inspection of the dataset will be used to classify the post type. For *celebrating* events, words which usually indicate special events such as birthdays and Christmas are used. For posts on *travelling*, synonyms as well as methods of traveling are used. For *eating*, aside from synonyms, the meals of the day are also used as indicators.

All post types stored in the *Verb Object* table are first sequenced according to date. For every classified post in the Verb Object table (Table 5.7), content determination constructs a message for each of it. Each constructed message

consists of the object from the Verb Object table pertaining to the said verb together with whom, when and where the event has happened.

With the sample event in *Verb Object* table (Table **??**), content determination can generate the message *write(Robee Khyra Te, thesis paper document, Janica Mae Lam, Camille Saavedra, Alds Hade, 11/22/2016, Gokongwei Building, De La Salle University - Manila).*

The final output of the content determination will be an abstract representation of the story plan that is composed of a set of messages or predicates that the system would like to convey to the reader. Each message in the story plan will follow the abstract representation of the form:

Verb(doer, receiver, object, date, location)

Given the story plan from content determination, sentence aggregation will then determine how the messages in the story plan can be combined to form a single sentence, as well as the relationship across two sentences based on rhetorical structure theory (Section 3.5.2.2 - Rhetorical Relations). Appropriate discourse markers will then be used, such as and, therefore, but, because, to name a few, in order to show if one sentence is used to explain, justify, elaborate or provide an example to another sentence. Pronoun generation will also be done in this process. Sentence Aggregation will also be responsible for determining the time elements of a message. Specifically, these time elements include "last <year/month/week>" "every <year/month/week>", "often", "recently", "<n> week(s)/month(s)/year(s) ago", and "every <n> day(s)/week(s)/month(s)/year(s)". For example, all messages reflecting events that happened in the past month will be aggregated together by the time element "last month".

Lastly, linguistic realization will be responsible for generating the surface form of the story text using SimpleNLG. SimpleNLG would also be used to automate orthography, morphology, and simple grammar verification.

GenConclusion will be used to generate the conclusion part of the story text, which will contain the user's likes and preferences. The exact process of determining the user's likes is explained in the Inferencing Section, 4.4.4.

The generated texts from the two text generation modules are then merged, and presented to the user as the complete life story. The user will now have the option to save or discard his/her life story.

5.2 Processing User-Generated Data

Facebook was chosen for this research for two reasons, namely: [1] its free-form nature; and [2] the amount of data present in Facebook. From a single Facebook post, plenty of information can already be derived in order to complete events that make up a life story, including, but not limited to: [1] date; [2] time; [3] location; [4] co-participants; [5] a photo, or photos, each of which could have their own separate post filled with their own metadata; and [6] the current activity being done by a person, if they chose to use the *predefined activities* feature.

This does not yet include the actual text content of a single post. From the text post itself, an intelligent machine can be designed to easily determine parts that can be used in natural language generation, such as the subject(s) of the post, the verbs, and the objects they act upon. However, user-generated data such as those from Facebook are inconsistent and noisy (Kinsella et al., 2011). In this section, each of these characteristics are discussed in detail and how the issues inherent in these characteristics are resolved.

5.2.1 Brevity of Posts

Includes: multi-sentence posts; and very brief posts with implied attributes such as actor, object, or time

User-generated data in social media is usually brief. Other social networking sites have limitations set in place such that user-generated data is brief on purpose. However, Facebook does not have a set character limit in place, which means that posts on Facebook can be much longer than others. This becomes an issue when a single post contains multiple sentences, each with their own actions and some with different doers. Other posts may be super short, which leads to missing attributes in the text such as the doer, the object, or time.

In dealing with this, long posts with multiple sentences are split into sentences and then parsed per sentence. During preprocessing, Stanford CoreNLP takes care of splitting such post into its constituent sentences, and classification is performed on the individual sentences.

Recall that the story plan is of the form

Verb (doer, receiver of the action, object, date, location)

For posts with missing elements in the text needed to fill the story plan, those

elements can be found in the metadata instead, or assumed.

- If the *doer* is not mentioned in a given sentence (e.g., "Had fun today!), the user who posted it is assumed to be the doer.
- If the *receiver* is not mentioned then the poster is assumed to be the receiver.
- The *object* can be missing.
- The *date* is taken from the post's metadata.
- The *location* can be missing; it is taken from the post's metadata.

5.2.2 Informal Nature of Posts

Includes: presence of unnecessary characters; presence of foreign characters; emoticons

Continuing to echo the findings of (Kinsella et al., 2011), posts on social media are more often than not informal, and there is a tendency to resort to hyperlinks or attachments for context. These characteristics were evident in our dataset, wherein it is hard to classify text posts because much of it is humor based around context which a computer cannot know. Also, posts containing foreign characters, emoticons, laughter and hashtags abound. During preprocessing, these were removed as they currently have no relevance to the classification and the generation tasks.

5.2.3 Parsing Sentences

Includes: POS tagging; multilingualism; parsing sentences with multiple verbs

Parsing sentences refers to breaking down a post into its different sentences and then breaking down the sentence into its parts and being able to describe their syntactic roles. To do this, POS tagging is done by Stanford CoreNLP. It generates a constituent and dependency representation. From this output, syntactic analysis is performed to extract the necessary event details (Manning et al., 2014).

Given the post "Going to the mall. Relevant elements are extracted following these steps:

- Extract the verbs that signify the activity described in the post, and the objects or the recipient of the action, which may be another person or object. In this example, the verb is "going and the object is the noun phrase describing the destination, "to the mall.
- Apply lemmatization to transform words to their lemma in order to increase the accuracy of the classifier. In this case, "going is lemmatized to "go.

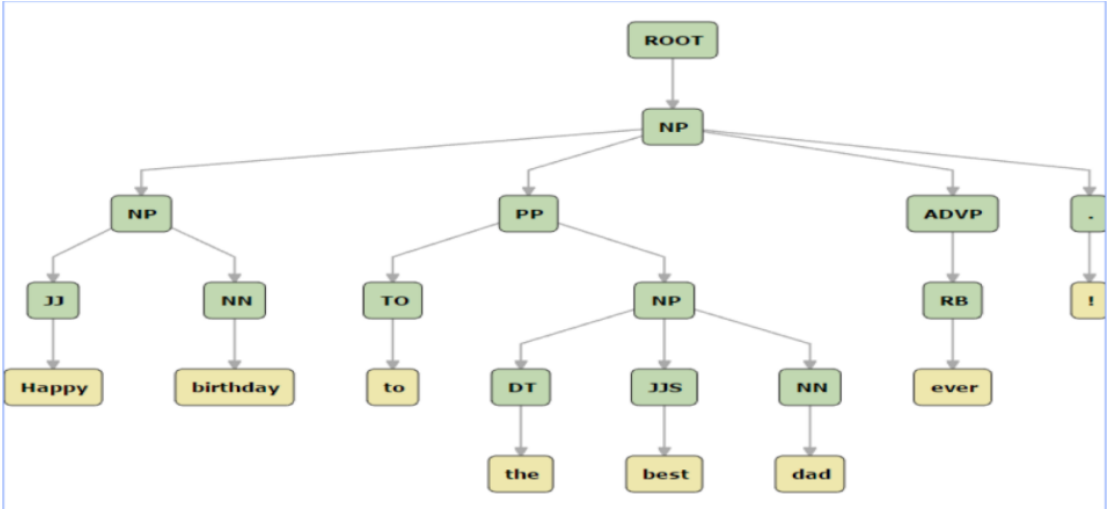

Figure 5.7: A sample parse tree generated using Stanford CoreNLP

However, the above steps do not work in all cases. Given a sample post – "*Happy Birthday to the best dad ever!*, Stanford CoreNLP generates the parse tree shown in Figure 5.7, which contains the tokens, their POS tags and their dependencies. The object is "*best dad*; however, the post does not contain any verbs. Posts with no verbs will have to rely on the event classification algorithm to determine its category, in this case, it is a *celebrating* post because of the keyword "birthday.

The accuracy of the POS tagging also relies on Stanford CoreNLP, which led to problems because Stanford CoreNLP is not perfect. Depending on the punctuations, for example, a sentence's POS tagging changes.

To illustrate, the post, "Happy friendversary thesismate HAHAHA, with a tagged person named "Camille Saavedra, shows up in the generated story as "Robee celebrated friendversary thesismate HAHAHA with Camille. This is because the POS tagger detected "friendversary thesismate HAHAHA as the main noun phrase in the post, and so considered it to be the object of the post. POS tagging can also change as a result of multilingual sentences; the use of mixed

languages mess up the syntactic structure of the post, making it difficult for the parser to properly perform POS tagging.

After breaking down the sentence and completing the story plan, it is possible to have multiple words in the sentences which leads to the post possibly being classified into multiple types. For example, the sentence, "Walking while eating gelato, contains keywords for both *travelling* and eating types of posts.

In the first iterations of the system, the post classification algorithm simply used co-location in order to check which types the post fell under. If it happened to have keywords for different types, the same sentence would appear multiple times in the generated story.

The post classification algorithm was changed: the keywords list was augmented with phrases from WordNet and ConceptNet; a scoring system has been implemented; and multiple post types are no longer possible. This change is further expanded on in Section 5.2, "Event Classification.

5.2.4 Other Identified Characters

Includes: Detecting rumors, contradictory information, sarcasms and humor, onomatopoeia, colloquial language

Aside from the aforementioned characteristics of social media data, there are plenty of other characteristics relevant to this field. The first is that a lot of information posted online is speculative and wishful. Being able to detect rumors and distinguish them from facts is therefore necessary; however, it is not part of this research. Another characteristic of many posts is humor and/or sarcasm; being able to detect sarcasm and humor is a concern of empathic computing and human-computer interaction, and is not part of this research. Both of these characteristics can affect event detection (because, for example, a sarcastic post can be misconstrued by a computer as fact), and they become more relevant in the future as the topic expands, due to the need to accurately portray users' lives. Humor and hearsay can be misconstrued by the computer as fact, and therefore, wrong information can be generated.

Other characteristics which are a result of informality on social media include onomatopoeia and colloquial language. Onomatopoeia is the formation of a word from the imitation of a sound associated with it (e.g. "oink oink, "buzz, or "haha). For this research, laughter is removed as part of preprocessing; however, other onomatopoeic sounds are not preprocessed. Colloquial language refers to the use of very informal terms that are not standard; for example, texting in

abbreviations (e.g. “c u l8r). For the purposes of this research, such language was not taken into consideration.

5.3 Event Classification

Although Facebook’s predefined activities feature is designed to enable users to easily classify their individual posts according to content, there are currently no available tools that can support the extraction of relevant elements from posts that use this feature. Furthermore, most Facebook users still prefer the traditional methods when crafting a post, i.e., typing text, and optionally combining photos and videos.

Given this limitation, available tools were used for gathering posts from an individual user’s Facebook account, preprocessing the posts, classifying posts according to their event types, and then extracting event details.

5.3.1 Keywords (Handcrafted)

During data gathering, the initial dataset containing 2,514 posts were collected from 16 Facebook users. These posts were manually classified based on what the researchers think was the correct classification. It is important to note that at first, there were 15 types of posts, each with their own co-locating words; later on, it was then trimmed down to four categories of posts. These categories were chosen based on the frequency count of the posts classified to be as events. These categories were: *celebrating, travelling, eating,* and *drinking.* Posts classified under these categories were analyzed and a reference table (shown in 5.8) containing the predefined keywords commonly associated with each event category was derived. Other posts not classified under these categories and no event posts were disregarded.

Table 5.8: Classification of Facebook Posts based on Keywords

Type of Post	Co-Locating Words
Celebrating	Birthday Celebrate Congratulations Congrats God bless Bless Wish Happy Merry Party
Traveling	Go Travel At Visit Drive Road Place Far Run Walk Adventure Bucket list
Eating	Cook Eat Dine Breakfast Lunch Dinner Chicken Burger Grill Bake Fry

Only posts under the four categories were analyzed to derive the keywords. For *celebrating* events, words which usually indicate special events such as *birthdays* and *Christmas* are used. For posts on *travelling*, synonyms as well as methods of traveling are used. For *eating*, aside from synonyms, the meals of the day are also used as indicators.

Since the dataset derived from manual inspection is by no means exhaustive of all Facebook user accounts, the list of keywords is not complete, and needed to be expanded to improve the classification process.

5.3.2 Keywords from Existing Resources (ConceptNet and WordNet)

In order to address the issue of lack of keywords, the combination of the outputs of the two lexical resources, WordNet and ConceptNet, were used to populate the keywords list to be used later for event classification (as shown in 5.11 and 5.12).

Initially, the plan was only to use WordNet only by extracting related concepts through its related senses. However, the lexical knowledge contained in WordNet was found to be insufficient for our purpose, as many terms (specifically physical objects) do not exist in it. To increase the coverage, ConceptNet's lexical and semantic knowledge was utilized to derive related contexts in which the words *celebrating*, *travelling*, *eating*, and *drinking* are found. Specifically, semantic relations such as "IsA, "MadeOf were used to derive the concepts. Table **??** shows the complete list of relations used and their descriptions. ConceptNet also contained a very minimal amount of Filipino words; these were also extracted to handle posts that contain Filipino words. An example Filipino word, *paglalakbay*, is shown in 5.12 as one of the keywords for the *travelling* post.

Table 5.9: List of semantic relations and their descriptions

Relationt	Description
HasFirstSubevent	What do you do first to accomplish it?
HasLastSubevent	What do you do last to accomplish it?
HasPrerequisite	What do you need to do first?
MadeOf	What is it made of?
IsA	What kind of thing is it?
AtLocation	Where would you find it?
UsedFor	What do you use it for?
CapableOf	What can it do?
MotivatedByGoal	Why would you do it?
Desires	What does it want?
DefinedAs	How do you define it?
InstanceOf	What type of thing is it a specific example of?
CausesDesire	What does it make you want to do?
Causes	What does it make happen?
HasSubevent	What do you do to accomplish it?
HasProperty	What properties does it have?
PartOf	What is it part of?
ReceivesAction	What can you do to it?
CreatedBy	How do you bring it into existence?

Table 5.10: Keywords Derived from ConceptNet (see Appendix M for full list)

Type of Post	Co-Locating Words
Celebrating	Victory Christmas Firework Birthday Toast
Eating	Chew Swallow Food Cook Plate Kain
Drinking	Thirsty Liquid Bottle Beer Lemonade Inumin
Traveling	Passport Fun Explore Pack Adventure Paglalakbay

Table 5.11: Keywords Derived from WordNet (see Appendix X for full list M)

Type of Post	Co-Locating Words
Celebrating	Celebrate Observe Fete Lionize
Eating	Eat Feed Depletion Consume
Drinking	Booze Salutation Pledge Salute Toast
Traveling	Movement Traveler Locomotion Journey Trip

Table 5.12: Keywords Derived from ConceptNet (see Appendix M for full list)

Type of Post	Co-Locating Words
Celebrating	Victory Christmas Firework Birthday Toast
Eating	Chew Swallow Food Cook Plate Kain
Drinking	Thirsty Liquid Bottle Beer Lemonade Inumin
Traveling	Passport Fun Explore Pack Adventure Paglalakbay

After integrating WordNet and ConceptNet, the system has 1,697 co-locating words across all event types. Table 5.13 shows a breakdown of the co-locating words.

Table 5.13: Number of the co-locating words per event type and source

Event Types	from WordNet	from ConceptNet	TOTAL
Celebrating	9	350	359
Eating	17	400	417
Drinking	24	492	516
Traveling	16	389	405
TOTAL	66	1631	1697

Initial testing was done to the keywords from ConceptNet and WordNet. However, the results were low due to the fact that the list of keywords were both broad

and vague. The list contains 1,697 keywords including repeating words that may overlap from one category to the other. When the classifier encountered this problem, it immediately classify this to the first category according to the hierarchy to be explained in Section 5.3.4 - Scoring System which may lead to misclassification. Words that are not closely related to the category were also included that causes the system to classify a post under that category when it should not be the case. Thus, the keyword list needs to be pruned hoping to attain a better result. Analysis was done to determine which keywords need to be removed. Removing excess keywords limits the outliers (keywords far in relation to the category) from affecting the classification of the posts. This concentrates the keywords to the words with the closest relation to the categories, thus improving the classification.

5.3.3 Pruned Keywords

Removing the excess keywords limits the outliers (or the keywords far in relation to the category) from affecting the classification of the posts. This concentrates the keywords to the words with the closest relation to the categories, thus improving the classification.

The set of derived keywords were pruned manually following a set of rules:

- Rules to retain a keyword:
  - One word verbs relating to the category which may or may not be followed by a noun (i.e. chew, swallow, eat meal in eating category; drink coffee,thirst, hydrate for drinking category; throw party for *celebrating* category; take bus, fly airplane and go sightseeing for *travelling* category)
  - Proper nouns pertaining to the action (i.e. coffee, tea, lemonade in drinking category, cookie and meal in eating category; cheer)
  - Modes of transportation, experience and verbs indicating travelling (i.e. take airplane, drive road, sightsee, jet lag for travelling)
  - Special events and holidays that may or may not come as greetings (i.e. anniversary, friendversary, Merry Christmas, Christmas Eve, New Year, wedding, party)
- Rules to remove a keyword:
  - Articles (i.e. a, an, the)
  - Common verbs (i.e. get, become and go)

- Repeating words (i.e. eat , cookie, eat cookie; remove eat cookie)
- Words that have no connection to the categories (i.e. esophagus, dance, call family, wet mouth, pee, fridge)
- Overlapping category (i.e. 'Champagne which is included both in drinking and celebrating; would be removed in celebrating since the action weighs more on drinking.)

After the pruning process, the system now has 521 co-locating words across all event types. Table **??** shows a breakdown of the co-locating words.

Table 5.14: Number of the co-locating words per event type and source after pruning process

Event Types	from WordNet	from ConceptNet	TOTAL
Celebrating	9	67	76
Eating	17	108	125
Drinking	24	197	221
Traveling	16	83	99
TOTAL	66	455	521

5.3.4 Naive-based System

The initial post classification algorithm used the handcrafted keywords list to classify the posts. The algorithm of the first implementation was: For each token in the sentence: For each keyword: If token matches keyword: Get keyword category Set category as post type

However, several issues were encountered. First, consider the post "Eating and drinking at McDo., the initial classifier would classify this post as both eating and drinking post. Later on, when generating the story, it will contain redundant sentences. Another issue found was that the classifier is very sensitive. For example, the post "At your service, was classified as a travelling post because of the keyword at. Because of this, the classification algorithm was improved to cater these issues.

5.3.5 Scoring System

As stated in 5.1.3, posts are classified by their relevant keywords. This is done by first breaking down the post into its individual sentences and getting the part-of-speech tags of each token, which is handled by Stanford CoreNLP. Afterwards, the verbs are lemmatized, and all relevant tokens in the sentence are cross-referenced with the table of classification stated in 5.2.1.

Because a single sentence can contain multiple verbs or words signifying events, the first iteration of the automated classifier classified this into multiple event categories. For example, the sentence, "*Walking around the streets of Rome while eating delicious gelato,* was classified as an eating event and a travelling event.

To avoid redundancy and the loss of context in downstream tasks, the classification algorithm has been revised to use a scoring system, and the sentence is assigned the category with the highest score. If multiple event categories bear the same score, a bias scheme based on the hierarchy of *celebrating => eating => drinking => travelling* will be followed. The hierarchy was based on the frequency count of the classified events using the gathered posts. A threshold value of 2 was also set to minimize the occurrence of misclassification. The threshold was set to 2 because the most common keywords contained at least two words. Most classified posts contain keywords such as "drink coffee, "eat food, "Happy Birthday, and "Merry Christmas. Setting the threshold to 1 increases the likelihood to be misclassified such as the previous example "At your service. However, increasing the threshold to 3 would then be limiting most of the posts. This would then result to under classification which means the value of false negative would increase.

Consider the sentence, "*I'd love to take a walk on the park someday.* The presence of the word walk in the list of keywords led the no-score classifier to consider this sentence as a travelling event, when it should not have been the case. In the score-based classifier, only sentences such as "*I'm going on an adventure to check off one from the bucket list*, which has a score of 3 because of the words "*going*, "*adventure*, and "*bucket list* and thus, was categorized as a *travelling* event.

The initial algorithm for the score-based classifier was: Set initial score to 0 for each category For each token in the sentence: For each keyword: If token matches keyword Get keyword category Increment score Set category with highest score as post type

But this new scoring algorithm still poses some issues. Using the example "At your service., the classifier matched the keyword "at so the category travelling has 1 point. The keywords for other categories no not match any of the words, so the

highest score was *travelling*. But the example post is not a *travelling* post, thus, the algorithm was tweaked to fix issues like this. The latest algorithm needs to satisfy at least 2 points for it to be classified as that category. The final algorithm was: Set initial score to 0 for each category For each token in the sentence: For each keyword: If token matches keyword Get keyword category Increment score If category with highest score ¿= threshold Set category with highest score as post type

5.4 Life Story Generation

The text generation module is responsible for generating the appropriate story segments that make up the entire life story generated by the system. There are three components of the text generation module, namely the GenIntro, GenBody, and GenConclusion. GenIntro is the module applied to the data determined to be facts or direct knowledge, such as family and work information; GenBody is applied to generate the body of the life story from the data processed in the Data Processing module; and GenConclusion is the module applied to the data determined to be facts or direct knowledge.

Following the NLG pipeline of Reiter & Dale (1997; 2000), story generation usually proceeds in three main steps, namely content determination, discourse planning, and surface realization. Initially, the system used a template-based generation algorithm, but later switched to a grammar-based (or script-based; the two terms are interchangeable in the context of our research) algorithm. A thorough discussion of the rationale behind the switch is explained in Section 5.3.6 Switching to Grammar-Based Generation.

5.4.1 Template-Based Generation

Following the NLG pipeline mentioned earlier, content determination for GenIntro and the GenConclusion involve deriving data supplied by the user themselves. The system initially used templates and was therefore *template-based*. The algorithm for GenIntro is as follows: For each subtemplate: Get all possible templates based on the available data Randomly choose 1 template Fill the template with the correct data Connect all phrases to form the whole introduction

For example, when generating the introduction, the system follows a general template of:

- <NAME> *<intro_birth>* *<intro_education_gs>* *<intro_education_gs>* *<intro_education_college>* *<intro_work>* *<location, hometown>* *<intro_family¿*

  Then, each of these has their own sub-templates. For instance, *<intro_birthday>* can choose from the following templates:

  - *<intro_birth_circumstance>*
  - Is a <AGE>-year old <GENDER> who

  Then, the *<intro_birth_circumstance>* can still be broken down to other sub-templates such as:

  - , born on <BIRTHDAY>,
  - was born on <BIRTHDAY>

  Finally, upon choosing the templates at random, the system would then fill in the blanks with the data stored in the *direct knowledge* table as follows:

  Robee, is a 21-year old female who has yet to get her college diploma from De La Salle University.

  While for the GenConclusion (Liked pages), the general template was: <NAME> likes <liked pages> and the algorithm is as follows: For each category: Use the template: ¡category¿ such as ¡pages¿ In generating the ¡pages¿ phrase: Check how many examples from that category. If yes, iterate through each one of them. Use ", to connect the first 2 then use "and to connect the last. Get the plural form of the category. If only 1, no need to determine connector. And no need to get plural form of the category. Replace ¡pages¿ to the generated phrase Replace ¡liked pages¿ to the generated phrase

  Lastly, for the GenConclusion (Attending Events), it uses the general template: ¡NAME¿ attended ¡events¿ and the algorithm is as follows:

  For each event: Use the template: ¡event name¿ at ¡location¿ In generating the ¡events¿ phrase: Check how many events If more than 1, iterate through each and one of them. Use , to connect the first few events then use and to connect the last. Check if there is a location for each event. Replace ¡events¿ to the generated phrase

Same algorithm was used in the GenConclusion - Interested Events but this time is uses the general template: <NAME> was interested in attending events such as <events>.

The system would then chain templates like these together in order to create the introductory and conclusion paragraphs. Following the NLG pipeline mentioned earlier, this would be discourse planning, but very minimal discourse planning is performed in a template-based approach due to how simple it is. Each time, it would have to choose a sentence at random. This poses a problem: the templates are manually created by a person, and so if a new sentence type were to be implemented, such as about a person's phone number, a template would have to be created, which includes all possible varieties of sentences that can be formed which talk about the phone number. Also, note that since templates are not flexible, it introduces problem for surface realization: there are usually grammar problems encountered such as the use of the correct pronoun or the correct preposition. These would then have to be worked around by the system, which means more work had to be done.

Therefore, a way to dynamically generate sentences using NLG is needed for extensibility and scalability. Works such as ((Chen, Lim, Perez, Reyes, & Lim, 2008); (Sleimi & Gardent, 2016)) have shown the power of using RDF data to construct descriptive sentences dynamically. This approach was adapted, which enabled support for graph-based, or grammar-based, text generation, using scripts instead of templates.

This meant that the story planning and the surface realization parts are changed, since the content determination stays the same.

5.4.2 Scripts

The story generator uses Resource Description Framework (RDF) data to construct sentences, with the help of SimpleNLG. RDF data consists of (*subject-predicate-object*) triples such as (*Robee, nationality, Taiwanese*). The subject indicates the resource, while the predicate indicates the trait or aspect of the resource which also shows the relationship between the subject and the object. As illustrated in Figure 5.8, RDF data can be represented by a graph in which edges are labelled with properties and vertices with subject and object resources.

Each vertex in the graph is an object (not to be confused with *direct object*), and for the purposes of story generation, each object is described in a sentence (turning it into an assertion or a message). Aside from the name of the object,

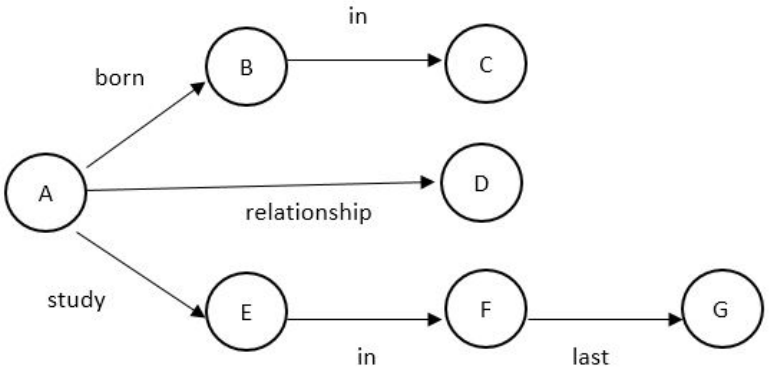

Figure 5.8: An example of a graph representation of RDF data. (A, birthPlace, F) is connected to (F, country, H), for example.

assertions can be a combination of the following.

For the introduction, the assertions are:

- person (lastName, firstName, middleName)
- gender(obj)
- livingIn(obj)
- family(relationship, names<>)
- roleFam(obj)
- occupation(obj)
- birth(date, place)
- education(institution, type, yeargrad, course)
- work(institution, startDate, endDate, location)

And for the conclusion:

- person (*lastName, firstName, middleName*)
- eventsGoing(*name, location*)

- eventInterested(*name, location*)
- likes(*category, page<>*)

These assertions are then filled with information from the user data, such that, for example,

person (lastName, firstName, middleName)

becomes

person (Hade, Alden Luc, Rosqueta)

A set of story grammar rules were used to form English sentences. Section 5.4.6, Switching to Grammar-Based Generation, discusses the reasons why the implementation of the Introduction and Conclusion shifted from Template-Based Generation to Grammar-Based Generation. This will be evident later when we discuss the evolution from templates to grammars in generating the Introduction (5.4.3), Conclusion (5.4.4), and Body (5.4.5).

The grammar rules used for the introduction are shown in 5.15; (5.15 is shown here as an example; the grammar rules for the conclusion and body are in their respective sections later on.)

Table 5.15: Grammar Rules Used for the Introductory Paragraphs

<INTRODUCTION>	<SENTENCE>+
<SENTENCE>	<subject> <PREDICATE>
<PREDICATE>	<verb> <OBJECT>
<OBJECT>	<noun> [<preposition phrase>*] || <article> <noun> [<preposition phrase>*] || <preposition phrase>

The system reads the grammar file using the bottom up approach, where it will start with filling the grammar rules at the bottom with data, before working its way up. Grammar rules that have not been filled up will be removed, while grammar rules that have been filled will be used to generate the assertion. The system then loops through the list of assertions for the introduction and conclusion, fills them with data, generates the sentences with the help of the grammar

rules, and puts the sentences together, in order to generate the paragraphs for the introduction and conclusion respectively.

An example for the introduction would be that the following assertions

person (lastName, firstName, middleName)

gender(gender)

livingIn(location)

with the help of data from Facebook would become

person (“Te”, “Robee Khyra”, “”)

gender(“Female”)

livingIn(“Manila, Philippines”)

and would generate the following RDF triples (note that the surface form of the verbs are defined as part of the RDF triples):

(“Robee Khyra Te” “is” “female”)

(“Robee Khyra Te” “lives in” “Manila, Philippines”)

Each of these RDF would then become a sentence of the form

<sentence> -> <subject> <predicate>

where

<predicate> -> <verb> <object>

Therefore, the generated sentences would become

Robee Khyra Te is female.

Robee Khyra Te lives in Manila, Philippines.

Putting the sentences next to each other would result in a paragraph.

5.4.3 Generating the Introduction

The introduction paragraphs are meant to present the Facebook user to the reader, to provide a background of the subject as if in a real biography.

Template-Based Generation Iteration 1
The very first iteration of the generation of introduction was done simply to check whether the data is being used correctly, and how well the templates would look when put together into a paragraph. The templates were simply filled up and concatenated with each other. Since each template was not a complete sentence by itself but rather a clause, the output paragraph was not separated into sentences. Also, templates which were not filled with data would glaringly have missing information. An example of this is shown below.

Robee Khyra Te was born **on 05/25/1996 got her high school diploma** from Chiang Kai Shek College **in 2013 graduated college** in De La Salle University **last 0 worked from** 2013-09-01 to 2014-04-30 at University Student Government, **DLSU is from <hometown>** she is the daughter of Ian Quintin and Katherine Ann Te.

Template-Based Generation Iteration 2 - Missing Information
For the first true iteration of GenIntro, these templates were put together into sentences with the help of SimpleNLG. However, it still could not account for missing information:

Robee Khyra Te was born on 05/25/1996. **She got his high school** diploma from Chiang Kai Shek College in 2013.
She got his college diploma from De La Salle University on <grad_year>.
She worked from **2013-09-01 to 2014-04-30** at University Student

Government, DLSU. She hailed from Manila, Philippines, and is now living in Manila, Philippines. **She she** is the daughter of Ian Quintin and Katherine Ann Te.

Worth noting is that the template in the database, since it was created by a human, assumed a lot of things about what the computer can do by itself (such as the simple issue of separating the paragraph into sentences, or determining the user's gender).

For the next iterations, the missing information were accounted for. If a template cannot be filled completely, clauses related to the missing information (such as the dependent clause "from <hometown>) are removed. The introduction text also still did not know if a student has graduated already or is still studying, for example. Also, the gender of the subject was not yet determined, so there are contradicting pronouns in the paragraph.

Template-Based Generation Iteration 2 - Missing Information

However, the text was still not completely readable. Dates were listed as they are in the database rather than written like natural language. And so, the generation was further improved. The gender of the user was eventually determined correctly; dates were made readable; and the system now took into account the years of education or work in order to determine whether they happened in the past or not, and then explain them in a way that makes sense. Another problem was redundant data: if the current city and hometown are the same, the same town appeared twice, perhaps even in the same sentence.

The last iteration before switching to grammar-based story generation produced an introduction which provides a concise, coherent, and grammatically-correct description of basic information about the subject's life.

Robee Khyra Te, born on **May 25, 1996**, got her **high school diploma** from Chiang Kai Shek College **last 2013**. She **has yet to get** her college diploma from De La Salle University. She worked from **September 01, 2013 to April 30, 2014** at University Student Government, DLSU. **She is from Manila, Philippines.** She is the daughter of Ian Quintin and Katherine Ann Te.

The grammar rules for the grammar-based story generation of the introductions are shown in 5.15.

Robee Khyra Te , born on May 25, 1996 , got her high school diploma from Chiang Kai Shek College last 2013. She has yet to get her college diploma from

De La Salle University. She worked from September 01 , 2013 to April 30 , 2014 at University Student Government, DLSU. She is from Manila, Philippines. She is the daughter of Ian Quintin and Katherine Ann Te.

Grammar-Based Generation Final Output

5.4.4 Generating the Conclusion

The conclusion is meant to summarize what was said about the user in the body of the life story by stating their likes. These likes are supported by giving examples of related Facebook pages that the user has Liked. But the conclusion also provides support to these likes by showing examples of events attended by the person.

Template-Based Generation Iteration 1

The very first iteration of the generation of conclusion was done simply to list down the pages that the user likes per category in a form of a paragraph. An example of this is shown below.

Robee Khyra Te likes Community such as Technology Impact Summit 2017, RVR COB Week 2017, Annyeong Oppa , Romance of the Three Kingdoms, Status: Speak Up, Oms Giving, The Border Collective, University Vision - Mission Week 2016, DLSU Hackercup 2015, Handog 2015, Jazzy's Accessories, DLSU - Manila Secret Files, SOLIDarity Against Abuse, Alive: Touch the Sky, Hero of D Day UP, CCS Month 2014: Conexus, CRYO, L E G A C Y, KPOP Concert Philippines, DLSU Administration, CSO Annual Recruitment Week 2014, LPEP 2K14, Sweetooth , Sims 4 and Millennia: Ignite the Revolution . Robee Khyra Te likes Artist such as Calleftgraphy , Park Shin Hye - PSH ? ? ??, Song Hye Kyo , Song Joong Ki ? ? , Daehan Minguk Manse, Ha Ji Won (?? ? ??), Nikki Co, Lee Sang Yoon - Turkey, Daniel John Ford E. Padilla, Ji Sung ? ? - ???? - ??, JYP Actors, Lee Bo Young, Kim Woo Bin, Jo Seung Woo ? ? , Lee Jong Suk ? ? ??, Lee Min Ho's World, Lee Minho (? ?), Lee Bo Young / ? ?? and Lee Bo - young

Template-Based Generation Iteration 2 - Limiting the list

The initial output did not have any limit as to how long it would be, and so all of the users likes and the events they attended were slammed into the paragraph. This was quickly corrected by limiting the conclusion paragraph(s) to three Liked pages per type.

Robee Khyra Te likes **Community** such as Technology Impact Summit 2017, RVR COB Week 2017, **A**nnyeong Oppa.
Robee Khyra Te likes Artist such as Calleftgraphy, Park Shin Hye-PSH ?? ??**, S**ong Hye Kyo.
Robee Khyra Te likes TV Show such as Cinderella and Four Knights, Moonlight Drawn by Clouds - Korean Drama, **D**escendants of the Sun.

Template-Based Generation Iteration 3 - Forms, Punctuation and Capitalization

The problems dealt with in the improvement of the conclusion paragraph(s) were all related to grammar and punctuation and capitalization errors. However, first, there was the need to teach the generator to use pronouns, because saying the full name in each sentence is redundant. Some of the simple sentences to form longer sentences were then combined, once the system was taught to use pronouns:

Robee Khyra Te **likes Communities** such as Technology Impact Summit 2017, RVR COB Week 2017 **, A**nnyeong Oppa **., Artists** such as Calleftgraphy, Park Shin Hye-PSH ?? ??**, S**ong Hye Kyo**., TV Shows** such as Cinderella and Four Knights, Moonlight Drawn by Clouds - Korean Drama**, D**escendants of the Sun.

Also, a problem with the category of the pages is evident. Notice that even though there are three examples to support a single category, the category name used was still written in its singular form. The initial algorithm for this was to create a manual surface realizer to correct the form of the category. if(lastLetter is y) Change last letter to ies; Else if(lastLetter is s) Append es; Else Append s;

However, in the latter part of the development, the developers utilizes SimpleNLG for getting the plural form of the category.

Finally, the events attended by the user were plugged into the conclusion.

Table 5.16: Grammar Rules Used for the Conclusion Paragraphs

<CONCLUSION>	<SENTENCE>+
<SENTENCE>	<subject> <verb> <PHRASES>
<PHRASES>	<SIMPLE_PHRASE> || <COMPLEX_PHRASE>
<SIMPLE_PHRASE>	<NOUN_PHRASE>+
<NOUN_PHRASE>	<noun> <LIST>
<LIST>	"such as (<noun> [<prepositional phrase>*])+
<COMPLEX_PHRASE>	<GERUND_PHRASE> <INFINITIVE_PHRASE> <LIST>
<GERUND_PHRASE>	in <verb>
<INFINITIVE_PHRASE>	to <noun>

Grammar-Based Generation Final Output

Robee Khyra Te likes communities such as Technology Impact Summit 2017, RVR COB Week 2017, Annyeong Oppa, artists such as Calleftgraphy, Park Shin Hye-PSH, Song Hye Kyo, TV shows such as Cinderella and Four Knights, Moonlight Drawn by Clouds - Korean Drama, Descendants of the Sun. Robee Khyra Te attended Cybersecurity and International Relations, Publication Writing Workshop, LSCS Christmas Party! at The Manila Residences Tower II in Manila, CCS Month 2016: Festivo at Henry Sy Bldg, De La Salle University - Manila and Technology Summit 2016 Forum at De La Salle University in Manila.

The grammar rules for the grammar-based generation approach for the conclusion paragraphs are shown in 5.15.

The outputs for grammar-based story generation for both introduction and conclusion are similar, with the most significant changes happening under the hood rather than on the surface (or the generated story, in this case).

5.4.5 Generating the Body

The body is meant to show events regarding *celebrating, eating, drinking,* and *travelling*, with the observation that these posts are most explicitly stated by

Facebook users. The bulk of the work for the story generator is in producing the text for the body of the life story. Following the NLG pipeline mentioned earlier, content determination involves utilizing the events that were derived from processing and classifying the posts. Story planning is then responsible for organizing and sequencing the events into a coherent story plan, which is comprised of sequences of events of the form

Verb (doer, receiver of the action, object, date, location)

In generating the story plan, the planner takes into consideration the temporal and the topical relations of events. Topical relations are used to generate paragraphs; one topic (or event category) equates to one paragraph. Within each paragraph, events are ordered based on their temporal relations, which are determined from the timestamps attached to each post and linked to the corresponding events.

Surface realization converts each verb entry in the story plan into a sentence to express the date(s) of occurrence, as well as the people and places involved in each event. The task involves defining the input specifications for each sentence to be generated. This includes setting the user and other people tagged as the actor or doer of the action, the particular action, the object, and the tense of the verb.

The algorithm used for generating the body paragraphs was:

Get all event posts Sort by category For each category: Sort by date For each post: Generate a sentence for it Add a summary sentence for each paragraph

The summary sentences for each categories were:

- Celebrating - With whom did the user celebrated the most
- Eating/Drinking - Where have you eaten
- Travelling - Where have you been locally and internationally

Grammar-Based Generation Iteration 1

Consider the Facebook posts including the extracted metadata shown in Table 5.17, all classified as celebrating. The corresponding story plan was:

celebrating(Mae, null, Angie, 10/03/14, null)
celebrating(Mae, null, Jamie, 02/07/17, null)
celebrating(Mae, null, thesismate, 02/14/17, null)
celebrating(Mae, null, null, 08/16/16, null)

and the resulting story text was:

Mae celebrated Angie. Mae celebrated thesismate with Cam. Mae celebrated with Shane in Manila, Philippines. Mae celebrated anniversary Jamie HAHA with Jamie.

Grammar-Based Generation: Iteration 2 - Added Date, Location, and Tagged Fri

In the early iterations of GenBody, each event leads to one sentence of the form "On ¡date¿, ¡actor¿ celebrated ¡activity¿ with ¡friend¿. But before that, the dates, location, and tagged were not even mentioned at all, and the activities identified were almost always wrong.

The problems encountered with this approach stem mostly from difficulty in parsing posts that are mostly informal in nature. Some posts are parsed incorrectly, leading to the wrong activity being articulated. For example, in the last sentence above, the identified activity from the post is "anniversary jamie.

On October 03, 2014, Mae celebrated Angie. On February 14, 2017, Mae celebrated thesismate with Cam. On August 16, 2016, Mae celebrated with Shane in Manila, Philippines. On February 07, 2017, Mae celebrated Jamie HAHA with Jamie.

Grammar-Based Generation: Iteration 3 - Topic Sentence and Temporal Relation

One solution to improve the coherency and reduce the redundancy in the generated text is to take advantage of the temporal relations among events by sorting them from most recent to oldest according to their timestamp. This task is handled by the story planner. The surface realizer then applies aggregation to group related events together, i.e., those with closer temporal relations or those with the same people involved. Closer temporal relations mean either the same date, the same month or the same year.

Mae has celebrated most with Shane, Jamie and Cam. They celebrate together. A year ago, she celebrated with Shane mid-July. A few months ago, she celebrated friendversary with Cam. A few months earlier, she celebrated anniversary jamie with Jamie.

The algorithm used for generating the temporal relations was: If (same year) if(same month) Random = recently, in recent times, in recent past, not long ago; Else if(month ¡=5) Random = a few months ago, during the past few months); Else Random = this year in ¡month¿, almost a year ago, many months ago, early this year; Else Append in ¡year¿; if(day ¿=1 day ¡=10) Random = early that month, during the start of the month, as the month starts, early ¡month¿ that year; Else if (day ¿=11 ay ¡=20) Random = in the middle of ¡month¿, mid-¡month¿, during ¡month¿; Else Random = before the end approaches ¡month¿, as the month of ¡month¿ ends, near the end of ¡month¿ in that year;

While, the anaphora generation was derived from checking the direct knowledge gender from the direct knowledge table. If the extracted gender from Facebook was “male, the pronouns used were “he, “him, and “his. While, if the gender was “female, the pronouns used were “she and “her.

Similar to GenIntro and GenConclusion, the GenBody has been modified to accept data of RDF triples, and thus the generation also becomes grammar-based. The grammar rules for this are shown in Table 5.18.

Table 5.17: Sample Facebook posts classified as *celebrating* posts, along with their metadata

Original Post	Metadata	
Happy 18th Angie!	date created	10/03/14
Happy anniversary Jamie HAHA	date created	02/07/17
	user tagged	Jamie
Happy friendversary thesis-mate!	date created	02/14/17
	user tagged	Cam
Party party!	date created	08/16/16
	user tagged	Shane
	location	Manila, Philippines

Table 5.18: Grammar Rules Used for the Body Paragraphs

<BODY>	<SENTENCE>+
<SENTENCE>	<SENTENCE_SPECIFIC> || <SENTENCE_SUMMARIZED>
<SENTENCE_SPECIFIC>	<time> <subject> <PREDICATE_SPECIFIC>
<PREDICATE_SPECIFIC>	<verb> <OBJECT_SPECIFIC>
<OBJECT_SPECIFIC>	"to" <noun> <LOCATION_SPECIFIC> <PEOPLE_TAGGED>
<LOCATION>	"at" <noun>
<PEOPLE_TAGGED>	"with" <noun>+
<SENTENCE_SUMMARIZED>	<subject> <PREDICATE_SUMMARIZED>
<PREDICATE_SUMMARIZED>	<verb> <OBJECT_SUMMARIZED>
<OBJECT_SUMMARIZED>	<noun> <PREP_PHRASE>
<PREP_PHRASE>	<PLACE_PHRASE> <CITY_PHRASE> || <LIST> || <PEOPLE_TAGGED>
<PLACE_PHRASE>	"to" <noun>+ || "at" <noun>+
<CITY_PHRASE>	"in" <noun>
<LIST>	"such as <noun>+

Worth noting is that for GenBody, two types of sentences can be formed. These can either be sentences talking about a specific post, or a generalization or summary of multiple posts. An example of a specific sentence would be, He went to the mall with Janica at Glorietta, while an example of a summary sentence would be, "He celebrated the most with Janica, Robee, Alden and Camille.

5.4.6 Switching to Grammar-Based Generation

The structure used for the paragraphs of the life stories generated in this research, as well as the switch from template-based generation to grammar-based generation (in the form of scripts), is in part based on the work of (Tuffield, Millard, & Shadbolt, 2006).

When modeling a story, it is important to take note of the structure of *fabula* items; in the case of the system, the fabula refers to story elements such as characters, objects, and events. The structure is enforced by grammar rules– and these grammar rules are often enforced by templates.

However, as discovered over the course of this research, templates are rigid and have to first be defined by developers every time a pattern needs to be created. This limits the flexibility and scalability of the system. For example, if there exists a template for a sentence which introduces simply the user's name, the developer would have to define each different variation that can be produced by following this template. For example:

- <NAME> <optional_clause> <current_job_or_education>
- <NAME> <optional_clause> <birth_circumstance> (and then each of those elements except for *NAME* would have to be created manually as well, since they are not terminals; similar to the example in Section 5.3.1 Template-Based Generation)

If an end-user wanted more variations, the developers would have to manually define new variations. Worse still, if an end-user wanted to define new sentence types not accounted for, the developers would have to manually define each sentence type (e.g. introducing a person's pet, or birthday, or relationship status), and manually create each *variation* within each sentence type. This results in additional overhead which could have been solved by using grammar rules instead and allowing tools such as SimpleNLG to focus on surface realization.

This is why the system switched to grammar-based story generation instead: easier definitions and an aim for greater flexibility and scalability. In this case, it is now possible to generate varying sentences for the paragraphs without having to manually define new templates: the focus then goes to the grammar and the script. As stated by (Tuffield, 2006), "ontologies built around existing narrative theory offer a powerful way to tackle this problem at a more pragmatic level, without encumbering end users with additional overheads of conceptualising explicit semantics.

Chapter 6

Results and Observations

Chapter 6 discusses the different testing methods done on the system. Furthermore, it also shows the evaluation results from the Facebook users and experts.

6.1 Objectives of Testing

The system underwent testing for the following reasons:

- It is necessary to test the system in order to determine that it works as expected (i.e. each module behaves properly, without bugs or glitches);
- To ensure that the system satisfies its end-users; and
- To know how to improve on the system (and how much improvement is necessary) based on user feedback.

To this end, the system underwent two types of testing: [1] black-box testing; and [2] user acceptance testing.

Black-box testing was done to verify if a particular function or module behaved and worked as expected. This type of testing focuses on detecting errors and problems that involves wrong extraction of data, pre-processing errors, incorrect event details extraction, classification errors, generation errors, database issues, and initialization and termination errors.

User acceptance tests were performed by the target or end users. This test ensures that the features and behavior of the system satisfy the users. Further-

more, the test aims to determine possible revisions with the system based on user feedback.

6.2 Black-box Testing

Black-box testing was done to verify if a particular function or module behaved and worked as expected. This section presents the discussion of the different test cases for each module in the system.

6.2.1 Extraction

Graph API is used to extract data from Facebook and the system is responsible for storing the extracted data accordingly. FB Stories requires the presence of user-generated data to be able to generate a story; therefore, the data extracted from Facebook must have retained its integrity. To ensure this, the data is cross-checked with Facebook via the Facebook ID attribute to make sure that the data from the users profile and posts, to the JSON objects, to the database, was not tampered with or corrupted.

This process of checking integrity was first applied to a small selection of data. At first, only a small portion of the users account data was extracted and stored in order for the developers to be able to manually check the veracity of each piece of information extracted. When the process was confirmed to be effective in maintaining integrity, it enabled the system to extract all the data needed for story generation. This process of manual checking also ensured that all data extracted could be used by the system in some way using our current system architecture.

6.2.2 Text Understanding Module

Pre-processing Regular expressions were used to search for specific patterns of text such as emoticons, *haha*, and non-alphanumeric characters like Hangul and Kanji. In order to check the correctness of the resulting output, the raw texts containing different special characters were fed to the system, and then manual inspection of the input and output texts were performed to see if it was able to remove the unnecessary characters.

Event Detail Extraction During testing, there were several issues found in

the system that could be traced back to Stanford CoreNLP. These included issues with text segmentation and POS tagging.

Below are the issues associated with text segmentation:

- When splitting text into sentences, Stanford CoreNLP is highly dependent on the use of periods, often associating it as the end of a sentence. Thus, when a text is in a form of a list, (e.g. “1. Hi 2. Hello) instead of splitting it into two parts “1. Hi being the first sentence and “2. Hello being the second, it splits it into three parts: “1., “Hi 2., and “Hello.
- Stanford CoreNLP is problematic with words that contain apostrophes. It splits contractions into two tokens in ways which occasionally pose a problem during POS tagging. For example, it can split up the contraction, “Im, by turning it into two words, “I, and “m.

The following issues were identified when assigning POS tags on words in a sentence:

- Periods and commas are treated like any other token, thus influencing the tag chosen. Given the sentence“Robee is drinking., the POS tagger correctly identifies the word *drinking* as a verb. However, after removing the period at the end, leaving the sentence now as “Robee is drinking, the tagger associates *drinking* as a noun.
- Since there are a lot of ambiguities in English (as there are in all other human languages), Stanfords POS tagger has difficulty in interpreting forms ending in ing as verbs, nouns, or adjectives. For example, “Robee is flagging an issue. *Flagging* is a common verb used by many modern Internet developers, but it is tagged as an adjective.
- The tagger incorrectly annotates pronouns such as *anyone* and *anybody* as nouns. (E.g. “Anyone could have done that.) results to the word *anyone* being identified as a noun.
- For words that are not in English, there are instances where the POS tagger is able to identify the verb or the parts correctly, and instances where it fails to do so. Given the text “Gumising kana., Stanford is able to correctly identify the word *gumising as a verb*. However, given the text “Kumain ng mansanas., *kumain* is identified to be as a noun instead of a verb. This is a problem when mixed language sentences are parsed, e.g. “Im okay *lang*.

With regards to lemmatization, the output of the lemmatizer depends on the part-of-speech tag assigned to the original word. For example, given the text "Currently painting., the POS tagger tags the word *painting* as a noun, when in fact it should be identified as a verb. As a result, the lemmatizer returns the lemma of the noun *painting* as *painting* instead of *paint*.

In instances where Stanford is unable to identify and obtain the necessary data – when no explicit verbs can be found in the text, the system relies on the classification algorithm to determine its category. Given the text, "Merry Christmas, the classifier classifies the text as a *celebrating* post because of the keyword "Christmas, thus, the verb to be used is "*celebrated*. However, for posts missing other details such as a direct object, it is found to be more challenging, since the sentence that is to be generated later on during story generation would be, in a sense, incomplete.

6.2.3 Post Classification Module

Much of the work done in the system was on the post classification module. It is necessary for FB Stories to be able to classify posts properly because the classifications are used in generating the life story later on, and having the wrong classification would lead to the story either being incoherent (not easily understandable) or incohesive (if it can be understood, it will sound bad because an event that was misclassified would end up in the wrong paragraph).

Section 5.2 mentions the improvements done to the keywords list, as well as the fact that the system initially did not have a scoring system and was augmented. Both were done with the aim of improving the results of the post classification module. Therefore, the system was tested with the following combinations:

- Non-scoring-based algorithm (or older version of the system) with old keywords;
- Scoring-based algorithm (or current version) with old keywords;
- Non-scoring-based algorithm (or older version of the system) with new keywords;
- Scoring-based algorithm (or current version) with new keywords.

The dataset was first subjected to manual labeling before being ran by the classifier. Our dataset composed of 21,412 posts, and out of those, 1,298 (or 6.06%)

posts were identified as being either *celebrating*, *traveling*, *eating*, or *drinking*. Broken down, there are 643 celebrating posts, 193 eating posts, 53 drinking posts, and 409 traveling posts. The complete dataset, including the posts with no events, is then fed to the automated classifiers, without scoring and with scoring, to assess their performance.

Table 6.1: Results of event classification with the current list of keywords

Metric	Older version (no scoring) - old keyword	Older version (no scoring) - new keyword	Current version (with scoring) - old keyword	Current version (with scoring) - new keyword
Precision	21.92%	9.58%	45.02%	10.60%
Recall	37.75%	55.16%	8.01%	26.96%
Accuracy	88.08%	65.72%	93.83%	81.78%

As shown in Figure 6.2, the no-score automated classifier using the old keywords list has a precision of 21.92% (the number of correctly classified event divided by the total number of classified events) and recall of 37.75% (the number of correctly classified events divided by the total number of actual events). The score-based classifier using the old keywords list, on the other hand, has a precision of 45.02% and recall of 8.01%. While instances of misclassified events have been reduced, the recall is drastically low for the score-based classifier because of the implementation of the threshold value.

A similar observation was made in the results of the two systems after feeding the new keyword list. For both the no-score based and score-based classifier, using the new keywords resulted for a lower precision and higher recall. The reason for this was that the new keywords list was taken directly from WordNet and ConceptNet, and these words were not checked for their relevance to the category. There were certain keywords present in the *travelling* category such as *businessman*, *scientists*, that is far related to the category, resulting in the decreased precision. On the other hand, the recall significantly increased, since there were more keywords used which resulted to more posts being classified correctly.

Table 6.2: Results of event classification broken down per event type This is for the new keyword list only

Event Type	Older version (no scoring)		Current version (with scoring)	
	Rrecision	Recall	Precision	Recall
Celebrating	45.47%	97.60%	65.47%	98.91%
Eating	38.545%	88.76%	30.77%	100.00%
Drinking	18.02%	83.78%	20.00%	80.00%
Travelling	8.56%	74.21%	8.47%	83.33%

Table NLP2 shows the performance of the no-score and score-based classifiers for each category of events. In the no-score classifier, celebrating events achieved the highest precision (45.47%) and recall (97.60%) among all event types. This is because events tagged as celebrating are more explicitly stated compared to the other types of events, as seen in posts such as "*Happy anniversary to my parents,* and, "*Merry Christmas!* Events under drinking and travelling have low precision because posts in these categories are usually implied through the use of proper nouns, such as the name of a drinking place or the food, instead of the actual action. An example is, "*Beach!*. Since the list of keywords does not contain any proper nouns, our two classifiers cannot tag sentences such as "*at Mt. Tremblant for today!* as travelling and "*enjoying my daily cup of Starbucks* as drinking.

In the score-based classifier, celebrating events still achieved good precision and recall values. The threshold did not affect the classification because most posts contained at least two of the celebrating keywords, such as "*happy* and "*birthday*. The 100% recall value in the category eating means all 193 posts on eating were correctly classified. Again, events under drinking and travelling have low precision following the same problems identified previously. Should the post be stated as "*drinking my daily cup of Starbucks coffee*, the threshold would have been met with the keywords "*drinking* and "*coffee*.

Table 6.3: Sample posts of their classifcations. *(NS no-score classifier; SB score-based classifier; Act actual classification)*

Post	NS	SB	Act
Happy birthday to my favorite sister!	Celebrating	Celebrating	Celebrating
Drinking tea on a Sunday morning	Drinking	Drinking	Drinking
Drinking Swiss Miss on a cold day.	Drinking	No event	Drinking
I'd love a good drive as an adventure	Travelling	Travelling	No event

Figure 6.3 shows some sample posts and their classifications. The first post is classified correctly, from the keywords "*happy* and "*birthday*. The second post is also correctly classified, from the keywords "*drinking* and "*tea*. The third post,

however, was misclassified by the score-based system because it has insufficient keywords to satisfy the threshold. The last post was misclassified by both classifiers due to the keywords "*drive* and "*adventure* which pertain to travelling. However, looking at the posts context, it is wishful, pertaining to something that did not actually happen as of the time of writing. Therefore, it should not be considered an actual event.

6.2.4 Text Generation Module

Template-Based Story generation systems rely on text generation modules to generate sentences and paragraphs befitting the story plan. In the first few iterations, a template-based system was used to generate sentences for the introduction and conclusion. The main template consists of the following categories: persons birth, education, working experience, family and location. If no information is found in a specific category, then, no sentence would be generated under it. Each category has multiple sets of sentence templates that only requires the insertion of information to be deemed as a complete sentence. To choose one template from the pool of templates, a randomizer is implemented which allows a variety of outputs per run.

Valuing the importance of integrity, information retrieved from the database to the story generation module are double-checked. Data inserted into the templates are simply backtracked from the database and user profile itself to ensure that the information are observed to be true. Some data needs to be converted or translated such as the date format to become more understandable and fitting for the story, but the integrity of these instances are still tested through manually checking the root source of information.

Several grammar issues were observed in the template-based system regarding the inappropriate usage of tenses for verbs, misuse of the subject-verb grammatical number (singular or plural), and improper and incorrect use of punctuations. Since templates are already preset formats for the story generations, the words used are predetermined based on assumptions. In addition, since the implemented rules for the template-based system were minimal, they sometimes resulted to grammatical errors. This was a problem which initially we dealt with by coding exceptions in the system; however, moving to a grammar-based system dealt with these problems.

Grammar-Based The template-based system for introduction and conclusion later on switched to a more flexible and adaptable method and turned into a script-based, or grammar-based, system. It follows a grammar rule rather than a fixed

predetermined set of sentences. Grammar rules and sentence structure weigh more in script-based systems compared to template-based ones.

The body script, on the other hand, depends per category but it contains similar grammar rules to the introduction and conclusion. For the body, two types of sentences can be formed: either sentences talking about a specific post, or a generalization or summary of multiple posts.

Grammatical errors that were present in the template-based system can also be observed in the script based system, but with the flexibility of script-based, changing the tense of the verbs, switching from singular to plural and the use of punctuations are made easier with SimpleNLG. This allowed the realizer to correct grammatical errors instead of manually defining the grammar in the template based.

Missing values also occurred whilst testing. Since Facebook users have different information stored in Facebook, some information are not provided while some are. This caused false assumptions regarding the sufficiency of data for all users allowing nullable and empty data be inserted into the sentence. This caused incomplete sentences to be generated.

The body also encountered redundancy of data. There are assertions where the object is one of the person tagged allowing the sentence to mention the person twice within the sentence. Other than people, location such as cities and name of places have also encountered this issue.

Sentence structure plays a vital role in a script-based system since the system relies on this. It follows rules set whether how the sentence would be presented. Data are input in sentence structures for testing and are manually checked whether the nouns, verbs, and prepositional phrases are all in place according to the rule.

6.3 End User Testing

End user testings were performed to know their thoughts on the content in the generated story and to identify the points that needs to be improved on. There were twelve gathered evaluators who will be evaluating the resulting story generated by the system. The evaluation form and criteria that the Facebook users will use consists of four parts, namely [1] language composition, [2] introduction, [3] body, and [4] conclusion.

All evaluators are Facebook users who are 18 years old and above. They

were briefed about the features of the system. After which, they logged in with their Facebook credentials and waited for the system to generate the story. After reading the story, they were asked to evaluate the system by answering the evaluation forms. Comments and suggestions for improvement were gathered for future improvement of the system.

The twelve respondents which varies from age and occupation were asked to evaluate the resulting story. There were 5 female respondents and 7 male respondents. The testing and evaluation were done individually and took place in De La Salle University.

6.3.1 Evaluation Forms

The evaluation form is divided into four sections, namely [1] language composition, [2] introduction, [3] body, and [4] conclusion. As mentioned in Section 3.7, this research, the primary component to be evaluated is the quality of the resulting life stories, in terms of completeness. This is the reason for coming up with the latter three parts of the evaluation form. In addition, the style of writing must be analyzed, mainly to check spelling, grammar, and coherence in the story, which resulted in the formation of the part for language composition.

6.3.2 Facebook Users Evaluation

Language Composition The criteria language composition focuses on the systems story generation modules which evaluates the resulting sentences sentence structure, grammar, and readability. This section contained most of the lower scores with average scores of 2.5 to 3.7 As seen in Figure 6.4.

The garnered average score shows the diversity of Facebook in terms of the use in mixed languages, it was difficult for the Stanford CoreNLPs POS tagger to identify the verbs befitting the object pertained to form complete sentences. Some problems present here includes posts having long sentences with multiple verbs like Happiness is really not bought but it is given by God and we should cherish these happy moments with our love ones and our father is one of them. Fathers are big blessing by God to us and I really thank God for giving me great father like him. Dad, thank you for everything. I maybe stubborn son but you were always there to teach me the rights and wrongs. Happy Birthday Dad, i hope you had great one! returned *bought* as the verb, which was true because it was the first verb present in the sentence and *moments* as the noun or object being

referred to which is incorrect.

Another problem is the mixture of upper and lower cases in some words like THank You to all! MAy God Bless You!!! which tagged bless as the verb and *May God* as the object. The grammar here was mixed up and it cannot realize the capitalization of the words. This caused the POS tagger to tag *May* as a proper noun because it is capitalized. This shows how dependent the POS tagged is to the data. The major problem identified in the outputs was the flow of each paragraph. This is caused by paragraphs which contains few sentences; some of which does not even make sense so it resulted to a sudden jump of thoughts. An example of this shows that the first paragraph talked about birthday celebrations and followed by a paragraph which contains only the sentence On May 27, 2010, He tied Lakers.. Afterwhich, it jumped to the next paragraph because it only tagged one sentence as a travelling post. Additional to this, the only sentence (actual sentence is Series tied at 2-2 for Lakers and PHX Suns ..... lets go Lakers lets go) present in the *travelling* paragraph was misclassified as *travelling* because he was cheering using the words go which is a keyword under the *travelling* category.

Table 6.4: Average Results in the Language Composition Section

Criteria	Average Score
Sentences are grammatically correct.	2.50
Usage of pronouns are correct.	3.17
Punctuations are properly used.	3.17
Capitalizations of proper nouns are correct.	3.58
There is no redundant information present in the life story.	2.92
Sentences within a paragraph have good flow (e.g. no sudden jump from talking about travelling to eating in the same paragraph).	2.58
Words used are understandable.	3.17
Words used are appropriate (or respectful).	3.5
The entire composition can be considered a life story or an autobiography.	2.83

Introduction Specific As seen in Figure 6.5, most of the scores are high because the introduction requires less analysis and understanding compared to

the body paragraph of the generated story. The correctness of information, flow of sequence and historical background received a lower score compared to the other criteria within the introduction specific as it includes temporal relations and user background and these information are more analyzed than fixed information like birthdays, gender and current locations.

Compared to the other sections (language composition, body and conclusion specific), Introduction specific has the highest average of scores. It were easier to generate the introduction since the assertions were simpler and hard facts were easily inserted to the grammar rules prepared.

Table 6.5: Average Results in the Introduction Specifics Section

Criteria	Average Score
The subjects birthday is correctly displayed.	4
If available from the subjects Facebook profile: • The information about the subjects education is correct and is sequenced from the oldest to the most recent. • The subjects employment history is stated. • Family members are introduced in the story and are correctly labeled. • The subjects status or relationship is stated.	3.58
The subjects contact information is NOT revealed.	4
Overall, the Introduction provides a clear background of the user, including his/her education, work (if applicable), and family members.	3.75

Body Specific Most of the low scores are in the body paragraphs where because of the text understanding, event classification and story generation modules

that are highly reliant to the tools used by the system. The low scores are mostly caused by the misclassification and distinction of categories between each paragraph. The average scores as seen in Figure 6.6 are not that low, but there are some criterias that were individually scored as ones (1s), mostly because of the incomplete ideas and misclassification of posts. An example of this was the generated sentence in the body paragraph On September 09, 2016, She looked kind. under the category travelling. The sentence that it was rooted from was When life brings you down, look at the sky and you'll see all kinds of beautiful things.

Table 6.6: Average Results in the Body Specific Section

Criteria	Average Score
Events mentioned in the story can be traced back to a post or multiple posts.	3.42
Events are correctly classified into travelling, eating, or celebrating.	3.00
There is a clear distinction of the focus of each paragraph, separating travels from dining and celebration events.	3.08
Events in a paragraph are sequenced based on their date of occurrence (from most recent to oldest events).	3.58
Correct events are generated in the story text.	3.33
The system identified the correct people tagged as part of the events post.	3.50
The system identified the correct location where the event happened and is consistent with the location on the post itself.	3.33
The dates when the events happened are consistent with the dates tagged on the posts themselves.	3.67
Other details of the events, e.g., objects, are generated when available.	3.50

Conclusion Specific Conclusion Specific Section has the best results among the four sections. The lowest score given was 2 under the first and last criteria as shown in Figure 6.7. The generated story mentioned and listed the liked pages

of the user but it does not necessarily mean that it specifies ones hobbies and interest, as such, lower scores were given. In relevance to this, the list of attended events do not have much relevance to the hobbies of interests stated. Some events are simply created for eating out or attending debuts.

Table 6.7: Average Results in the Conclusion Specific Section

Criteria	Average Score
The story describes the users hobbies and interests.	3.75
The users hobbies and interests are expanded upon by including examples of the users likes.	3.75
The story denotes events that the user is interested in.	3.83
The story identifies several events that the user attended which are relevant to his/her hobbies or interests.	3.83

"Melissa Chan is born on May 22, 1995. She's living in Davao City. She studied at Davao Christian High School, studied at De La Salle University and studying at Davao Christian High School. Mother is Ellie Chan.
She likes Communities such as Bagshoppe master, Engineers of La Salle and DLSU Crushes, Medias such as Bright Side, National Geographic and WBP We Blog Ph and App Pages such as The Price Is Right Community, Cafe Life and Hotel City. She attended events such as Regina @ 18 in Glass Garden in Pasig and BBQ Night in Krus na Ligas, UP Diliman Quezon City in Quezon City."

The generated story of the anonymized evaluator above garnered the highest score over all other generated stories because there was no missing information, all data extracted from Facebook was properly used, and it does not contain a body paragraph. The body is the most problematic part of the story because the system needs to fully analyze, tag, and classify each posts to be able to generate a good story. In this case, the score was high because it only uses facts from the extracted data and no further processing was made.

Stephanie Reyes is born on September 19, 1996. She's living in Quezon City, Philippines. She studied at St. Stephen's High School, Manila last 2013 and studying at De La Salle University. Brother is Heinson J. Reyes. Sister is Christly Pagola, Kara Ko, Kimberly Pavon, Alyette Ang, 'Jaylica Anne Tan, Aryll Dy, Janica Mae Lam, Trisha Mae W. Pablo, Jenea Yu, Azalea Lee and Eizel Tan.
On September 20, 2016, She celebrated you. On February 19, 2015, celebrated

their new year together. On February 19, 2015, celebrated their new year together. On January 17, 2015, She celebrated birthday this gal !!!. On October 17, 2014, She celebrated birthday guama. On October 13, 2014, She celebrated birthday Francesm Flores !!. On September 21, 2014, She celebrated friends these birthday surprises. On July 06, 2014, She celebrated birthday Joey Timothy Jao !!!.
On June 19, 2017, She went it.
On June 19, 2017, She explored LSCS. On June 19, 2017, She made LSCS thinking. On September 09, 2016, She looked kind. On May 23, 2016, She went this place these gals. On December 24, 2016, She walked @flores. On November 24, 2014, She got ticket.
She celebrated the most with Heinson J. Tan, Danielle Abuan, Rose Daryl Abuan, Jean Benard Zach Abuan, Dylan E Jao, Ivan Floyd Flores and Jeiyo Jao.
She likes Communities such as DLSU University Vision-Mission Week, Sports on Facebook and CCS Month 2016: Festivo, Artists such as Jake Vargas, Pekoiman and Calleftgraphy and Musicians such as Pedro the Pianist, ELF's SJ Fanclub and EXO-M. She attended events such as LEAP 2K17, SLC 2017 Dinner in Canton Road Restaurant, Shangrila Fort in Taguig and Huling Hirit 2017 in Amphitheater, De La Salle University.

The lowest result garnered is from the anonymized evaluator under the name Stephanie Reyes. The low score was caused by the lack of cohesion and coherence within the paragraphs and misclassification of other posts. In this case, most of her posts are promotional like inviting friends for a particular activity and sentimental such as sharing insights or rants. Most of her posts does not contain any check ins, tagged friends or other information that could help the system get more information about the tagged events so the idea of the sentences are incomplete. Promotional and organizational posts are also misclassified as events because words used are categorized under keywords that the event classification module uses. The flow within the story is not clear as well since there are paragraphs are composed of incomplete sentences.

6.3.3 Traceability Evaluation

Further analysis were conducted on the generated life stories of the 12 test participants. Specifically, the contents of the story were traced back to the Facebook data. This section will not tackle all comments and findings gathered, only the major issues will be shown.

Finding # 1 - Wrong Relations in Family From one of the evaluator, problems were shown in generating his family members. Figure 6.8 shows the

extracted family information of the evaluator.

Table 6.8: Family Relationship Stored in the Database

Criteria	Average Score	
Emsky Lam	mother	10154944365392731
Kathleen Marinas	sister	10209225247813117
Imelda Dy	aunt	1585251351491614
Janica Mae Lam	sister	10207575653011620
Wilson Martinez	uncle	10154858631848467
Eden Martinez	aunt	10154097531481105
Maria Camille Ng	sister	10212260896216481

The expected output of the given data must be:

Mother is Emsky Lam. Sister is Kathleen Marinas, Janica Lam and Maria Camille Ng.

But the actual result was:

Brother is Emsky Lam. **Brother** is Kathleen Marinas, Janica Mae Lam and Maria Camille Ng.

The reason for this was that the relationship types was not change to its corresponding types stored in the database. The relationship type was hardcoded to brother when doing the black-box testing and was forgotten to change it later on during the actual testing.

Finding # 2 - Missing Data Another finding found was that his start and end date in his works were not specified in his Facebook data. Thus, the output for this was:

He worked at Bdo Private Bank during, to, worked at Business Management Society during, to, and worked at De La Salle University during, to,.

Another case of missing data was observed in the location of the user. Some Facebook users does not include their location or hometown. Such problem causes the output of the system to be:

He's living in.

The system forgot to check if the date started, date ended, location, and hometown were present. It always assumed that the information was available from the data stored in the database.

Finding # 3 - Wrong extraction of event details To be able to generate a sensible sentence, the important event details must be extracted correctly. But since the system is highly dependent on the results from Stanford CoreNLPs POS tagger, some of these event details were extracted incorrectly; thus, resulting to issues.

Majority of the issues here exists in the body paragraphs. The body paragraph is generated based on the available event details given to the event model. If there are incomplete event details, then, the resulting sentence will also be incomplete. Same issues are tackled when there are too much details, the system would only extract on the first verb and object relating to the verb seen. The event details mainly failed to extract the correct object.

Chapter 7

Conclusions and Recommendations

This chapter presents the overall assessment on the development of the system, FB Stories. It discusses how the general and specific objectives of the research were met, the issues currently encountered by the system and recommendations for the future development of this and other relevant systems.

7.1 Conclusions

The main goal of the research was to develop an application that generates one's story using data collected from his/her Facebook account. To accomplish this objective, we did the following:

- Defined a life story and its elements;
- Reviewed Facebook data, identified the data that can be derived and the available methods to gather the data;
- Characterized user-generated text content to develop algorithms that can work with the noisy data;
- Built a knowledge base to store the extracted data;
- Designed the algorithms for event detection, event classification, event details extraction, and story generation; and

- Validated the system using a set of evaluation metrics.

FB Stories is able to generate life stories using natural language processing and natural language generation techniques. It is able to utilize user data such as their profile information and preferences. It can detect and classify posts regarding celebrating events; eating and drinking meals; and travelling across places. The system supports posts written in, and writes posts in, the English language. It generates life stories with the use of a story grammar and scripts, which makes the design of this module more scalable. There is an abstract representation of the story plan that is composed of a set of messages that the system would like to convey to the reader. Each message in the story plan follows the abstract representation of the form:

Verb(doer, object, tagged, date, location).

FB Stories is able to achieve the following specific software objectives as stated in Section 4.2.2:

- Extract the needed data from Facebook based on permissions and filters.

  The system was able to sift through the data present in a user's Facebook account and extract the [1] data that can be used as is, such as the user's birthday, Facebook Events, and likes; and the [2] data which have to be processed since knowledge taken from them has to be processed first via parsing, classification, and sequencing.

  Throughout this process, ethical considerations that may arise from the use of personal data were taken into account. This was done via the use of an informed consent form and an orientation for participants prior to undertaking the study. Participants were clearly informed of the procedure as well as any risks that may occur, and how the researchers deal with these risks, especially in terms of protecting the confidentiality of their data.

- Use data processing techniques to analyze the input.

  Facebook posts are hard to deal with because user-generated data are informal, brief, and noisy. To deal with these, the collected posts have to be preprocessed to remove foreign characters, emoticons, hashtags, laughter; and for posts with missing doers, the poster is assumed to be the assumed

doer. A thorough discussion of the data preprocessing and text understanding techniques used to analyze user-generated data and represent them in an abstract story representation are found in Sections 4.2, Software Objectives, and 5.1, Processing User-Generated Data. The posts are simplified into clauses, then subsequently fed to the text understanding module, which utilized Stanford CoreNLP to extract event details that are needed by the story generator.

- Classify each post according to its type.

  The preprocessed text underwent classification as described in Sections 4.3.3, Post Classification, as well as in Section 5.2, Event Classification. Multiple versions of the classification algorithm were designed. At first, it was a simple algorithm which simply checked if a given sentence contained one or more words in the post classification table of keywords (the old version of which is shown in Table X in Section 5.2.1, Keywords).

  Later on, the algorithm was improved, which consisted of augmenting the table of keywords with knowledge from ConceptNet and WordNet as well as adding a scoring system with a minimum threshold of 2 and an enforced bias between events. The best performing algorithm among these versions, which is the latter version, was then selected. The classification of a post is stored as part of its event model.

  The classification hinges largely on the textual content of the post. We encountered posts that rarely provide sufficient data from which useful details about the event can be extracted. These posed numerous challenges to our classifier, which despite achieving a reasonable accuracy of 88.08% and 93.83% for no-score and score-based approaches, respectively, also had low precision and recall values (shown in Table NLP1 in Section 6.2.3, Post Classification Module).

- Use text generation techniques to generate a story.

  From the initial template-based story generation approach, a story grammar or script has been defined in order to be able to more easily generate varying sentences for the paragraphs without having to manually define new templates: the focus then goes to the grammar and the script. There are three sets of grammars; one for the introduction paragraph, another for the body

paragraphs, and another for the conclusion paragraph. These grammars are used to determine what to put in a given sentence, and the script dictates how the sentence types should be organized or sequenced in general. The information that was then stored and processed earlier is modeled in the abstract story representation in the form of frames that are easily accessible to the system, and used to generate the story.

- Allow users to save the generated stories into a text file.

  After generating the complete story, the system provides a Save function which allows the user to store the story into their own system, allowing them to make use of it later on.

  Our system can generate a complete story about a user assuming the user has at least the following minimum data:

  - **Name.** This is required by Facebook, except for the middle name.
  - **Gender.** Like the name, this is required by Facebook. Aside from "male or "female, the user can actually enter a custom gender. However, Facebook still forces the user to choose a pronoun, which allows our story generator to work either way.
  - **At least one educational background OR work history.** Facebook does not require this; however, this information is necessary to generate a complete story according to (Youse, 2005).
  - **At least one of: current location (currently living in), and hometown.** Facebook does not require this.
  - **Birthdate.** Facebook requires this: the month, date, and year. From here, the age can be derived easily.
  - **At least two posts pertaining to an action.** The user must have at least two posts talking about: [1] travelling, [2] celebrating, [3] eating, OR [4] drinking. Note that this does not include posts which use only predefined activities (e.g. "Robee Khyra Te is *feeling sad.*); these two posts should have a caption entered by the user (e.g. "Happy monthsary!). These posts already have metadata such as the date posted, or co-participants (if any other users are tagged), so it is not necessary to ask the user for the dates when, or the locations where, the posts occurred.

- **Liked pages.** At least two. They may or may not be under the same category.
- **At least one Facebook Event,** where the user said "Attending or "Interested.

The following are optional, but help:

- **Posts with activities we can classify; Liked pages; and Facebook Events.** The more posts with activities posted; the more pages Liked; and the more Facebook Events attended or interested in, the better.
- **Family members.** These can be used for the introduction, but are not required by Facebook to be entered by the user. These allow the system to provide more information in the introduction; however, not everyone has at least one family member using Facebook.
- **Check-ins.** These help the story generator system tell the location of events that occurred in the users life.
- **Tagged friends.** These let the story generator tell who the user was doing their activity / activities with.

From these, it can be concluded that Facebook provides enough data to generate a complete life story about a person.

FB Stories is composed of different modules and works with the help of different tools for these modules. Because of the type of data the research deals with and the scope of the system, several issues were encountered. The following are the analyses for the components of the system, detailing how they affect the performance of FB Stories.

Graph API and Extraction

Extracted data from Graph API gives the system the raw data. Each piece of data extracted from Facebook retains its integrity because the system cross-checks its Facebook ID. However, as mentioned earlier, the extracted data is noisy, which means that it needs to be preprocessed to remove unnecessary characters and symbols, and classified properly to become information that can be used by the system.

Part-of-Speech Tagger

Some of the results of the tagger were different from what was expected. During testing, several issues were identified, which included not being able to distinguish some pronouns correctly; difficulty in interpreting verbs that end with *ing* due to ambiguities in English; and the high impact of punctuations such as commas and

periods in changing the result of the tagger when assigning POS tags on words in a sentence.

Lemmatizer

Stanford's lemmatizer did not have a problem in retrieving the lemmatized form of the verb. The only issue was when the part-of-speech tagger incorrectly identified the supposed verb as a noun or other part-of-speech tag, as it is highly dependent on the result of the tagger.

WordNet and ConceptNet

In constructing the keywords list, related words and concepts of the words *celebrating*, *eating*, *drinking*, and *travelling* were derived from WordNet and ConceptNet. A total of 1,697 related words were retrieved – 1,691 words are in English and only 6 are in Filipino. It can be argued from the results that some words taken from WordNet and ConceptNet negatively affected the precision of post classification. This is because these words were not checked for their relevance to the category. There were certain keywords present in the *travelling* category such as *businessman*, *scientists*, which are not closely related to the category. On the other hand, the addition of these keywords significantly increased recall, since there were more keywords used which resulted to more posts being classified correctly.

Dealing with Noisy User-Generated Data

Posts extracted and gathered from Facebook have to be preprocessed to clean the data of unnecessary symbols and non-alphanumeric characters, unstructured posts that contains incomplete posts, and a variety of quotes, words and phrases. This is a challenge for any research dealing with user-generated data.

GenIntro and GenConclusion

The introduction paragraphs are meant to present the Facebook user to the reader, to provide a background as if in a real biography. In a real biography, the following elements should be present toward the beginning: [1] birthday, and birthplace; [2] family members; and [3] childhood and education /citeYouse2005. Each of these pieces of information can be easily obtained from a Facebook user's About Me section. The only part here that is not described in detail is *childhood*; however, some writers circumvent this lack of information by describing parents, order of siblings, and the family's hometown.

The conclusion paragraphs, meanwhile, are meant to synthesize the user's experiences that were displayed in the body section of the biography by summarizing their preferences and interests. This is present in order for the reader to be able to understand the subject better. For the conclusion paragraphs, the goals were to present [1] hobbies and interests (Youse, 2005), as well as [2] events that they have attended. Both of these are obtainable from Facebook.

From here it can be seen that a Facebook profile, provided that the user is somewhat active, can provide enough information to generate textual data about the user which introduces him or her to other people.

GenBody
The body paragraphs, meanwhile, are meant to talk about events regarding *celebrating*, *eating*, *drinking*, and *travelling*, with the observation that these posts are most explicitly stated by Facebook users. However, in the dataset for this research, only 6% were tagged as either of these four events. While that may seem low, it is important to note that these are only four types of events, while there are plenty of other types of events posted on social networking sites, such as posts about *watching*, *listening*, or *playing a game*.

There is also a lot of data on social networking sites such as Facebook which do not talk about a person's activity, but are simply attempts at humor or spreading news to other people. Therefore, while it can be observed that data from Facebook is noisy, it can also be concluded that this *noise* leaves room for improvement in the future for other systems to take advantage of.

SimpleNLG
SimpleNLG was utilized in the story generation module. Initially, templates were used rather than natural language generation, which resulted in a "fill-in-the-blanks style of story generation with lots of grammar errors which had to be worked around in the system, since the templates are rigid. However, since changing to grammar-based story generation, these grammar errors can be fixed much more easily with the help of SimpleNLG. No issues were found with the tool itself as it was able to construct sentences properly based on what was fed into it.

7.2 Recommendations

The research was able to achieve its objectives, and FB Stories was able to perform its necessary functions. However, several areas could be improved to increase the effectiveness of the system. The following describes some recommendations for future work:

Python
Java and Python are both powerful programming languages. The big difference between them is that Python is dynamically typed while Java has to deal with the overhead of static types. This allows names to be bound to objects at execution time which makes it more convenient to use and call assertions in the system and carry out information without having the need to pass everything from one

function to another.

Python also supports more APIs geared towards ML processes compared to Java as well as APIs which distinguish between languages in a sentence. This would help identify sentences containing multiple languages and further improve the text processing module.

Use Machine Learning for Event Classification

Machine learning shows its significance when the system is exposed to new data. With machine learning, a system could adapt independently and learn from previous knowledge and be able to produce reliable results. The current system's event classification could benefit from using machine learning techniques because Facebook posts are highly subjective to the individual user, and also dynamically change over time. No single set of predefined rules will be able to capture this current and future variability. This way the system also would not be limited with the set keywords.

Utilize Facebook's Predefined Activities feature

Facebook was chosen for its freeform nature and the many ways a user could post information. This includes the ability to predefine what the user is doing at the time of posting, which helps out immensely in researches such as life event detection in social media. However, no method currently exists (as of the time of writing) for extracting predefined activities from posts in Facebook; this is the main motivation for post classification in our research. In the future, being able to utilize Facebook's Predefined Activities feature, through either first-party or third-party data extraction tools, would enable the system to work faster for certain posts (since the system no longer has to classify posts with a predefined classification already).

In addition, this *would* enable the system to more easily extend its support for other types of posts beyond *celebrating*, *eating*, *drinking*, and *travelling* by being able to quickly classify posts as *watching*, for example. When combined with machine learning, the system could theoretically learn new post types by itself without having to be trained, so long as that post type is one of the predefined activities of Facebook.

Story generation will then arguably be improved as more activity types are introduced. It would allow for a greater variety of possible sentences, which, if combined with other recommendations for NLG, would arguably make the story look more natural. The grammar can handle any new activity types easily because they were not made with only four types of posts in mind (e.g. none of the verbs are specifically stated in the grammar and can thus be *anything*); they were made with thoroughness and traceability in mind (the user must be able to trace any

given sentence in the body paragraphs back to one or more Facebook posts).

Improve the natural language generation

The story generator uses RDF data to construct descriptive sentences. However, the current implementation does not take advantage of RDF because the grammar only generates one type of sentence (e.g. for the birthday, the system can only write, "Robee Khyra Te was born on May 25, 1996). Therefore the full potential of RDF to construct varied, dynamic sentences was not realized. Further works can therefore improve on story generation by improving the use of RDF.

The current implementation of the story generator makes use of temporal and topical relations that exist between events, but is constrained to a select set of topic categories, as well as temporal rule sets to add cohesion and coherence to the sentences. Events should not only be limited to such categories but to more relevant events such as those identified by (Choudhury & Alani, 2014b) as being the most important: [1] graduation, [2] marriage, [3] new job, [4] having a newborn child in the family, and [5] undergoing surgery. Of note is that in Choudhury & Alani's (2014) research, they also used keywords in order to find tweets related to the topics they want to find (e.g. marriage "wedding, "tied the knot).

In addition, causal relations between events can also further enhance the story generation as it increases the value of context within the story. For example, it can show that a successful thesis defense is the cause of a happy graduation; or that a holiday (a holiday present in the calendar such as Christmas) is the result of a traveling trip (which happened on Christmas), which makes the story more dynamic.

Investigate the use of sentiment analysis to improve post classification and story generation

Sentiment analysis is a field of study that analyzes people's opinions, attitudes, evaluations, and emotions towards different topics, from people, to issues, to products, to events, among others (?, ?). It is a powerful tool to describe the emotions conveyed by a piece of text, and more broadly, the attitude or mood of an entire conversation (Varol, Ferrara, Davis, Menczer, & Flammini, 2017). Adding sentiment analysis to the topic enables the system to determine which posts signify the user's most emotional moments in life, which may help in determining the most interesting content. These may be accomplishments such as graduation or marriage, or setbacks such as a broken tire in the middle of a road trip.

Combining sentiment analysis with the system's post classification algorithm will enable the system to determine which content should be included in the final story (since it can eliminate posts perceived as boring or inconsequential). It can also allow the body paragraphs to be organized in terms of emotional intensity in

addition to the different types of events that happen in the user's lives (and this could, for example, allow the life story to have a sort of *climax* event written before the conclusion, which signifies the most significant moment in this user's life so far). Put simply, sentiment analysis may help improve the content determination of the system by enabling it to work with emotions; also, it may help the story generation become more interesting by further improving the flow of the story.

Using phrasal tagging

Currently, in POS tagging, each word (separated by spaces) is tagged separately. This causes problems for when phrases such as "watch out!, wherein instead of *watch* being tagged as a verb and out being tagged as a particle, the two tokens are tagged as a noun and a preposition respectively. A mistake like this can change the entire meaning of the sentence.

Our keywords list also contains phrases, such as "Happy mother's day for *celebrating*. However, when the keywords are searched in a given text, such as, "So happy my mother made me soup on this rainy day, the system finds the words *happy*, *mother*, and *day*, and thus equates this to "Happy mother's day.

A solution to these problems is phrasal tagging, wherein the system tags phrases like "Happy mother's day only for instances wherein the sentence actually says, as is, "Happy mother's day. It would remove the misclassification errors specified earlier.

Being More Sensitive to Context: Classifying Implicit Facebook Posts

Many posts on social media do not have explicit actions stated, being reliant on context that not everyone has. In our current classification algorithm, these posts, instead of having been classified in the correct category, are tagged as being not events. Examples are posts of eating wherein only the name of the restaurant or a picture of the food being eaten is posted; or instead of drinking, only the brand of the drink is stated. To this end, future researchers can look into things such as the following:

- Using metadata from hyperlinks (e.g. YouTube videos have a title, description, category, and miscellaneous tags) in order to help get context for a post; (Kinsella et al., 2011)
- Being able to augment the knowledge base with additional real-world knowledge such as relevant nouns (e.g. "Starbucks -¿ "coffee -¿ "drinking), so that text posts or metadata containing those nouns are not simply ignored; or
- Use of a gazetteer, in order for places, food, or other relevant proper nouns to be easily recognized by the named entity recognizer;

- Being able to use image processing techniques to find out the presence of objects such as food, airplane tickets, or drinks in order to be able to find objects related to activities such as *eating*, *drinking*, or *travelling*.

These will aid in post classification and content determination for the generated story.

Investigate Different Event Types

The results of the system showed misclassification errors because the post categories have some overlap with each other. For example, in *celebrating* an event such as a birthday, one might *drink* and *eat*. Further researchers can investigate different event types for classification and see whether more Facebook categories get included into the life story, as well as whether these classifications are more accurate and result in a more comprehensive life story.

Addressing Redundant User Posts

Unlike Twitter, which checks if a tweet is identical to another from the same user, a Facebook post can be identical to another post, thus allowing redundant posts (with the only difference being the *fbID* attribute). In our current system, this phenomenon could generate multiple identical sentences (if the redundant posts all have events). In order to solve this, checks can be added in the *to_be_processed* table to remove redundant posts.

Creating Personas from Generated Facebook Posts

One of the uses of life story generation from Facebook is for software agents to be able to use information from Facebook data to make sense of a person's activities and experiences. Assuming the ethical issues are taken care of, one of the possible uses for this is for defining personas. These are imaginary people which characterize the archetypical users of a system [1], conceived to show an example of the kind of people for whom a system is designed. They are meant to be based off of knowledge of real users, and Facebook provides plenty of real-world knowledge. To this end, work can be undertaken to determine if it is possible to create personas from life stories generated by a computer.

[1] Ambler. http://www.agilemodeling.com/artifacts/personas.htm

Appendix A

Resource Persons

Ms. Ethel Chua Joy Ong
Adviser
College of Computer Studies
De La Salle University-Manila
ethel.ong@delasalle.ph

Mr. Genaro R. Gojo-Cruz
Assistant Professor Lecturer College of Liberal Arts
De La Salle University-Manila
genaro.gojo-cruz@dlsu.edu.ph

Ms. Maria Clara M. Pacis
Assistant Professor Lecturer College of Liberal Arts
De La Salle University-Manila
ma.carla.pacis@dlsu.edu.ph

Appendix B

Similarity Report

This appendix contains the similarity result.

Turnitin Originality Report

THSST3 by Robee Te

From THSST Document (THSST)

Processed on 2017年07月12日 15:04 PHT
ID: 830397042
Word Count: 47791

Similarity Index	Similarity by Source	
12%	Internet Sources:	6%
	Publications:	3%
	Student Papers:	8%

sources:

1 2% match (student papers from 20-Mar-2017)
Submitted to De La Salle University - Manila on 2017-03-20

2 1% match (student papers from 14-Dec-2016)
Submitted to De La Salle University - Manila on 2016-12-14

3 1% match (student papers from 08-Dec-2016)
Submitted to De La Salle University - Manila on 2016-12-08

4 1% match (student papers from 12-Dec-2016)
Submitted to De La Salle University - Manila on 2016-12-12

5 < 1% match (Internet from 02-Mar-2015)
http://duepublico.uni-duisburg-essen.de/servlets/DerivateServlet/Derivate-38143/Ganster_Diss.pdf

6 < 1% match (student papers from 12-Dec-2016)
Submitted to De La Salle University - Manila on 2016-12-12

7 < 1% match (Internet from 21-Apr-2010)
http://www.clt.mq.edu.au/~rdale/publications/papers/1997/jnle97.pdf

8 < 1% match (student papers from 28-Jul-2016)
Submitted to iGroup on 2016-07-28

9 < 1% match (Internet from 31-May-2016)
http://www.accessecon.com/pubs/CEDWP/default.aspx?ID=20532&page=ViewDirectory

10 < 1% match (student papers from 07-Dec-2016)
Submitted to De La Salle University - Manila on 2016-12-07

11 < 1% match (Internet from 21-Sep-2016)
http://besttechmagazine.com/facebooks-artificial-intelligence-research-lab-releases-open-source-fasttext-on-github/

12 < 1% match (Internet from 11-Apr-2017)
https://en.wikipedia.org/wiki/Cyc

13 < 1% match (Internet from 23-Mar-2016)
http://aclweb.org/anthology/Y/Y08/Y08-1013.pdf

14 < 1% match (Internet from 11-Apr-2010)
http://www.calvin.edu/~kvlinden/distributions/vanderlinden-nlg-draft-1999.pdf

15 < 1% match (student papers from 28-May-2012)
Submitted to University of Newcastle upon Tyne on 2012-05-28

16 < 1% match (Internet from 25-Dec-2016)
http://scholarworks.uni.edu/cgi/viewcontent.cgi?article=1093&context=hpt

17 < 1% match (Internet from 10-Oct-2012)
http://nlp.stanford.edu/software/corenlp.shtml

Appendix C

Mr. Genaro Gojo-Cruz Interview Transcript

Date: September 29, 2016
Time: 2:30pm-2:50pm
Interviewer: Janica Mae Lam & Robee Khyra Mae Te

Introduction of Thesis Topic before actual interview

Robee: Ano yung mga common types of stories na usually ginagamit or sinusulat

Mr. Gojo-Cruz: Nagiiba iba eh.. depende sa mga issues ngayon, diba? Usually mga personal essays na maiikli na pang, ano lang, pang status kasi sinasabing wala nang time magbasa ang mga tao ngayon ng mga mahahaba. So yung status nya maiikli lang tapos depende yung theme usually kapag kung anong yung uso na topic, yun yung nagiging theme ng topic ng post.

Robee: ano yung mga common naman na elements? Kasi po focus po namin is life stories so gusto po namin malaman yung mga elements needed.

Mr. Gojo-Cruz: So tawag diyan, ano, personal essay or creative non-fiction. Yan yung tunguhin pala ninyo, ang tawag diyan ay creative non-fiction so ito yung batay sa totoong pangyayari, batay sa totoong experiences. Kung ito ay batay sa totoong pangyayari, laging "I", the one narrating my own story, so they use "I" as a pronoun. Your the subject mismo.

Janica: Sa mga creative non-fiction or personal essay may mga required ba na

elements? Like akilangan ba may plot palagi?

Mr. Gojo-Cruz: Wala, wala naman. Basta 100 hindi naman 100.. Kung magagawang 100% na kwento mo ng totoo yung akranasan yun yung creative non-fiction

Janica: So more on events yung creative non-fiction?

Mr. Gojo-Cruz: Kasama dun pero more on personal experiences na hindi lahat nakakadanas kaya ikwenekwento mo kasi... kakaiba. Kwinekwento mo kasi hindi siya common.

Janica: So different po siya sa biography?

Mr. Gojo-Cruz: Ang biography kasi nakafocus sa achievements kasi dito pwede mo ikwento yung, halimbawa, nagkaSTD ka kung pano ka naghanap ng ostiptal na magtretreat sayo na hindi ka kilala yung mga ganon, yung personal essay. So sa biography kasi puro highlights lang eh, diba? Ng buhay mo kwinenkwento mo mga rugs to riches, mga ganun, sa non-fic hindi ganun.

Robee: May mga common na structure ba yung mga elements, kunyai eto kailangan palaging nafofollow after yung ganitong element.

Mr. Gojo-Cruz: Ang nababuti sa ano creative non fiction malaya ka walang structure. Bsta sa huli lagi kang may iiiwan sa reader mo.

Janica: Like parang conclusion?

Mr. Gojo-Cruz: Conclusion pero pag conclusion kasi amsyadong academic ang dating eh. Ano iiiwan mo sa reader anong point of view anong philosophy, anong pananaw, realization, anong reflection na hindi laging aral.

Robee: Pano mo magagawang parang mamotivate yung audience niyo para matapos yung story?

Mr. Gojo-Cruz: Ako ang technique ko ano eh, pagnagsasalita ako sa essay ko hindi ako perpekto. Like as it is kwekwento ko, kung nakakahiya, wala akong paki. Kasi layunin mo talaga ay ma-ishare yung karanasan mo , kakahiya man ito or magmumukha kang sabaw, kasi ang problema sa mga writers, na ano, parang ang talitalino nila, napakaperpekto nila, so ang aking pagsulat hindi ganun. Laging may loop holes, minsan palpak kasi kapag nagkakamali ka mas nagiging human ka sa iyong readers, hindi laging ikaw matalino ikaw laging may alam.

Janica: may pre process po ba kayo kunyari before magsulat may ginagawa

po ba kayo or finoformulate?

Mr. Gojo-Cruz: Depende eh minsan may mga essays na pagdating ko dito sulat na kaagad kasi nadaanan ko mga gnun. Nadaananko sa taft mga ganun pero minsan halos wala naman ako nasusulat so hindi ko naman pinipilit. Kung walan gisusulat edi walang problema

Robee: Nakapagsulat na ba kayo gn stories para sa adults or teenagers?

Mr. Gojo-Cruz: Meron akong isang libro na Connect the dots, young adults ko. Collection of creative non fiction.

Robee: For our last question, pano niyo nirereview yung stories niyo, pano mo nalalaman na yung naproduce is okay na, naintindihan na ng reader mo or enough na?

Mr. Gojo-Cruz: Sa ano FB status, kung ipopost mo ito. Syempre mababa lang naman batayan nito kasi madali lang naman ilike ito. Syempre yung ano, yung number of naglike, ang pinakamaganda sa akin yung ano eh, yung number ng shares. Kasi kapag nagshare sila, lalagyan nila ng sarili nilang words kung paano nakapekto sa kanila yung story so yun yung number of nagshare nung post.

Appendix D

Ms. Maria Clara Pacis Interview Transcript

Date: October 12, 2016
Time: 11:20am-11:32am
Interviewer: Camille Saavedra

Introduction of Thesis Topic before actual interview

Camille: What are the common types of stories?

Ms. Pacis: In Facebook?

Camille: Yes, or in general.

Ms. Pacis: Im not a Facebook user. What do you mean? Life stories, autobiography, profiles. Those are the usual. But maybe for Facebook, it would just be profiles. Meaning, not birth to death, just a particular moment, event, time, or a time period.

Camille: What are the common elements of stories used in writing life stories?

Ms. Pacis: All the elements of fiction. You can use. Plot, character, dialog

Camille: Are all of these always required?

Ms. Pacis: Yes. We don't have maybe that dialog. Definitely for your characters to be more alive, they will have to be speaking.

Camille: What would you consider to be the essential parts of a story?

Ms. Pacis: All of that. It also depends on what you want to focus on. So for example, okay, do you already have an idea who you want to target for your stories?

Camille: Just any Facebook user.

Ms. Pacis: No, you have to choose. It cannot just be any Facebook user. If all the Facebook users says I woke up this morning and brushed my teeth. I mean, you definitely wont have a story. But let's say you choose Kim Kardashian, that's different right? What of Kim Kardashian is so fascinating? Maybe her obsession with her body or something like that? That's what you focus on. Okay? So, you have to look for a subject that is interesting. Or has something to say. Or have done something interesting. I would encourage you to choose themes or subject matter or people who have done good things. Very productive to help others. How many are you going to do ba?

Camille: We're not sure yet.

Ms. Pacis: Well, you can have a person like Kim Kardashian and contrast with someone who does good. So it's obvious to the reader who you should be emulating.

Camille: What is the distinction between a regular story and an autobiography?

Ms. Pacis: A life story is based on fact. Regular stories, there are fiction which is imagine or made up.

Camille: What are some common structures of stories that you notice?

Ms. Pacis: Did you guys take HUMALIT? You are asking me very basic questions.

Ms. Pacis: There's beginning, middle, end, And well it depends on how you guys want to write your stories. If you want to write stories and make it sound fiction, then you got to have a conflict. There should be a problem. Or it could just be a straight up narrative.

Camille: Have you written stories that target teenager or adult readers?

Ms. Pacis: Many. Yeah.

Camille: What are some of these stories you've written?

Ms. Pacis: Just look for it na lang at National Book Store. But its more for the younger not college. 12 year olds.

Camille: How do you keep the audience motivated to finish reading the story?

Ms. Pacis: You just have to keep the story exciting. But if you're using Facebook stories for your data then not very long.

Camille: How would you know if a story is good or how would you evaluate a story?

Ms. Pacis: If it's well written, if its captivating, or if it keeps you interested.

Appendix E

Student Research Ethics Clearance Form

This appendix contains the student research ethics clearance form document.

RESEARCH ETHICS CLEARANCE FORM
For Thesis Proposals[1]

Names of student researcher/s:	Hade, Alden Luc R. Lam, Janica Mae M. Saavedra, Camille Alexis T.R. Te, Robee Khyra Mae J.
College:	College of Computer Studies
Department:	Software Technology Department
Course:	BS Computer Science with specialization in Software Technology
Expected duration of project:	from: September 2016 to: August 2017

Ethical considerations

Human Participants

To the best of our knowledge, the ethical issues listed above have been addressed in the research.

Ms. Ethel Ong
Name and signature of adviser/mentor
Date:

Ms. Charibeth Cheng
Name and signature of panelist
Date:

Ms. Nathalie Lim-Cheng
Name and signature of panelist
Date:

Noted by:

Dr. Rafael Cabredo
Name and signature of department chairperson
Date:

[1]The same form can be used for the reports of completed projects. The appropriate heading need only be used.

Appendix F

General Research Ethics Checklist

This appendix contains the general research ethics checklist document.

DE LA SALLE UNIVERSITY
General Research Ethics Checklist

This checklist is to ensure that the research conducted by the faculty members and students of De La Salle University is carried out according to the guiding principles outlined in the Code of Research Ethics of the University. The investigator is advised to refer to the De La Salle University Code of Research Ethics and Guide to Responsible Conduct of Research before completing this checklist. Statements pertinent to ethical issues in research should be addressed below. The checklist will help the researchers and evaluators determine whether procedures should be undertaken during the course of the research to maintain ethical standards. The University's Guide to the Responsible Conduct of Research provides details on these appropriate procedures.

Details of the Research	
Students	Alden Luc R. Hade Janica Mae M. Lam Camille Alexis T.R. Saavedra Robee Khyra Mae J. Te
Thesis Adviser	Ms. Ethel Chua Joy Ong
Department	Software Technology
Title of the Research	Generating Life Stories From Facebook Posts
Term(s) and Academic year in which research is to be conducted	Terms 1 to 3 of A.Y.2016-2017

This checklist must be completed AFTER the* De La Salle University Code of Ethics *has been read and BEFORE gathering data.

Questions	**Yes**	**No**
1. Does your research involve human participants (this includes new data gathered or using pre-existing data)? If your answer is **yes**, please answer **Checklist A (Human Participants).**	YES	
2. Does your research involve animals (non-human subjects)? If your answer is **yes**, please answer **Checklist B (Animal Subjects).**		NO
3. Does your research involve Wildlife? If your answer is **yes**, please answer **Checklist C (Wildlife)**.		NO
4. Does your research involve microorganisms that are infectious, disease causing or harmful to health? If your answer is **yes**, please answer **Checklist D (Infectious Agents).**		NO

5. Does your research involve toxic/chemicals/ substances/materials? If your answer is **yes**, please answer **Checklist E (Toxic Agents).**		NO

Research with Ethical Issues to address:

If you have a YES answer to any of the above categories, you will be required to complete a detailed checklist for that particular category. A YES answer does not mean the disapproval of your research proposal. By providing you with a more detailed checklist, we ensure that the ethical concerns are identified so these can be addressed in adherence to the University Code of Ethics.

Declaration of Conflict of Interest

[/] I do not have a conflict of interest in any form (personal, financial, proprietary, or professional) with the sponsor/grant-giving organization, the study, the co-investigators/personnel, or the site.

[ ] I have a personal/family or professional interest in the results of the study (family members who are co-proponents or personnel in the study, membership in relevant professional associations/organizations).

Please describe the personal/family or professional interest: _____

_____

[ ] I have propriety interest vested in this proposal (with the intent to apply for a patent, trademark, copyright, or license)

Please describe propriety interest: _____

_____

[ ] I have significant financial interest vested in this proposal (remuneration that exceeds P250,000.00 each year or equity interest in the form of stock, stock options or other ownership interests).

Please describe financial interest: _____

_____

Declaration

We certify that we have read and understand the De La Salle University Code for the Responsible Conduct of Research and will abide by the ethical principles in this document. We will submit a final report of the proposed study to the DLSU-Research Ethics Office. We will not commence with data collection until we receive an ethics review approval from the College Research Ethics Committee.

_____ _____

Name and Signature of Student 1 Name and Signature of Student 2

_____ _____

Name and Signature of Student 3 Name and Signature of Student 4

Endorsement from thesis adviser to the thesis panel for proposal defense…	
Name and Signature of Adviser	Date

Endorsement from thesis adviser to the thesis panel for final defense… This is to certify that the research was conducted in a manner that adheres to ethical research standards. I am thus endorsing the group for final defense.	
Name and Signature of Adviser	Date

Appendix B

Certificate of Ethical Clearance

Appendix G

Research Ethics Checklist for Investigations involving Human Participants

This appendix contains the research ethics checklist for investigations involving human participants document.

DE LA SALLE UNIVERSITY

Checklist A

Research Ethics Checklist for Investigations involving Human Participants

This checklist must be completed AFTER the* De La Salle University Code of Research Ethics and Guide to Responsible Conduct of Research *has been read and BEFORE gathering data. The University Code of Research Ethics is available at http://www.dlsu.edu.ph/offices/urco/forms/URCO-Code-of-Research-Ethics_August2011.pdf

NOTE: This checklist is completed after the research proponent fills out the General Checklist Form.

Only answer this Checklist if you answered YES on question 1 of the General Checklist.

Researcher Details	
Students	Hade, Alden Luc R. Lam, Janica Mae M. Saavedra, Camille Alexis T.R. Te, Robee Khyra Mae J.
Thesis Adviser	Ms. Ethel Chua Joy Ong
Department	Software Technology Department
Title of the Research	Generating Life Story From Facebook Posts
Term(s) and Academic year in which research is to be conducted	Terms 1 to 3 of A.Y. 2016-2017

Provide a brief description of the data collection procedure to be undertaken in the research:
A Facebook user will be logging in his Facebook account. Information such as one's personal background, likes, events, family, education, works, and posts will be extracted and stored in the database.

The following should be attached to the checklist:
• A copy of the informed consent form to be used in the study. • A copy of the instrument/tool that will be administered to the participants. • If applicable, a copy of the letter seeking permission to collect data from participants who are under the supervision of an agency, institution, department, or office. • If applicable, a copy of the parental consent form for participants below 18 years old.

The following items refer to important ethical considerations in the conduct of research with human participants. Provide a check for the appropriate answer to each question.

Source of data *Please check all that apply:*		
	1. New data will be collected from human participants If you checked this item, how will the new data be gathered? Please check all that apply. **After answering this question, please proceed to page 3**	
		Experimental Procedures/Intervention/ Treatments
		Focus Group
		Personal Interviews
		Self-administered Questionnaire
		Researcher-administered Questionnaire
	/	**Internet survey**
		Observation
		Telephone survey
		Others, please specify:
	2. Pre-existing data from human participants, i.e., from a dataset **If you checked this item, please proceed to page 7**	

If both options are checked (both new data and pre-existing data), **answer all of the questions in this document.**

ONLY ANSWER IF NEW DATA WILL BE COLLECTED (item 1 above)

Sampling Details	
Number of Participants/Subjects	**12**
Location where the participants will be recruited/ where subjects will be obtained?	**De La Salle University**
How long will the data collection	**15 – 20 minutes**

take place?	
Who will perform the data collection?	**Researchers and the Facebook User**
Location(s) where data collection will take place	**De La Salle University**
What procedures will be employed to ensure voluntary consent from participants?	**Orientation of the process** **Briefing and accepting the terms written in the informed consent.**
Data Retention	
How long will data with participant identifiers be kept after the publication of the first paper from the project?	**1 year**
How long will anonymized data be kept after the publication of the first paper from the project?	**1 year**
Procedure for Informed Consent	
How will informed consent be recorded? (check all that applies) Reminder: please attach informed consent that will be used in the study	[/] Written Consent [ ] Audio-recorded Consent [ ] Online/Email recorded Consent [ ] Others, please specify:

If you will not obtain a recorded informed consent, answer the questions that follow:

Why does the waiver of informed consent not pose a threat to the welfare and rights of the participants?

Why is recording an informed consent not practical for the proposed study?			
	Yes	**No**	**Not Applicable**
1. Will the research involve students who will be receiving course credits for their participation? **If YES, please attach a copy of the consent form and a**			

summary of the debriefing process that will help participants understand how their participation in the research has provided a relevant learning experience to the crediting course.			
2. Does the study involve participants below 18 years old or those who are unable to give their informed consent? **If YES, please attach a copy of the parental consent form.**			
3. Is there a possibility that the research can induce physical and/or psychological harm to the participants? Will they experience pain or some discomfort as a result from their participation in the research? **If YES, please attach an acceptable argument that outlines the benefits of doing the research and how they outweigh the cost of harming the participants.**			
4. Will the participants be deliberately falsely informed or made unaware that they are being observed? Will they be misled in a way that they will possibly object to or show unease when told of the real purpose of the study? **If YES, please attach an acceptable argument that outlines the benefits of doing the research and how they outweigh the cost of harming the participants.**			
5. Will the research involve the discussion of, or questions on, sensitive topics (e.g. sexual activity, substance abuse, or mental health)? **If YES, please make sure that the informed consent form explicitly states that sensitive questions will be posed and that you will safeguard the anonymity of the participants and ensure confidentiality. Please attach a copy of your informed consent form and your instrument.**			
6. Will the research involve the administration of drugs, or			

other substances to the participants? **If YES, please attach an acceptable argument that outlines the benefits of doing the research and how they outweigh the cost of harming the participants.** **Please also attach a description of the procedure that will ensure that the participants will be brought back to their physical and psychological states prior to their participation in the research.**			
7. Will biological samples (e.g. blood, saliva, urine) be obtained from the participants? **If YES, will this involve invasive procedures? Please attach a description of these procedures.**			
8. Will genetic materials be obtained from the biological samples? **If YES, please attach a description of the procedures that will ensure confidentiality. Please attach the informed consent form.**			
9. Will financial inducements (other than reasonable expenses, like transportation or meal allowances) be offered to the participants for their participation in their research? **If YES, the researcher(s) should be mindful of how the inducements can influence the participants' responses or behaviors during the research. Indicate the financial inducements offered to the participants:** _____			
10. Is there a possibility for groups or communities to be harmed by the dissemination of the research findings? **If YES, please attach a description of procedures to ensure the anonymity and confidentiality of the research findings.**			

Answering <u>YES</u> to most of the above items will signal an ethical issue that needs to be addressed. Some actions that will allow adherence to research ethical principles are provided with each item. The researcher is advised to refer to the University's Guide to the Responsible Conduct of Research for the appropriate procedures to ensure adherence to ethical principles in the conduct of research.

Declaration

We certify that we have read and understand the De La Salle University Code for the Responsible Conduct of Research and will abide by the ethical principles in this document. We will submit a final report of the proposed study to the DLSU-Research Ethics Office. We will not commence with data collection until we receive an ethics review approval from the College Research Ethics Committee.

_____
Name and Signature of Student 1

_____
Name and Signature of Student 2

_____
Name and Signature of Student 3

_____
Name and Signature of Student 4

Endorsement from thesis adviser to the thesis panel for proposal defense…

_____
Name and Signature of Adviser

_____
Date

Endorsement from thesis adviser to the thesis panel for final defense…

This is to certify that the research was conducted in a manner that adheres to ethical research standards. I am thus endorsing the group for final defense.

_____
Name and Signature of Adviser

_____
Date

FOR PROPONENTS WHO WILL GATHER NEW DATA ONLY, PLEASE STOP ANSWERING.

Use of Pre-existing Data collected from Human Participants		
Indicate the dataset from which the data for the study will be sourced		
Is the data publicly available, i.e., the access to which does not necessitate an approval process?		Yes Please indicate where the dataset is available:
		No Please indicate/attach the approval authority for access:
Was the original dataset originally collected for the present study's purpose?		Yes **Please attach the Consent Form used in the original study.**
		No **Please attach the Information Collection Statement (i.e., the statement given to informants providing them with the rationale for the collection of specific information).**
Does the original data set contain sensitive data, that is information that an individual would not likely want to be disclosed publicly, e.g., data on sexual activities, substance use?		Yes **Please describe the type of sensitive data to be used in the present research:**
		No
Does the original dataset have personal identifiers?		No *(This means that neither the researcher nor the participant provided any personal identifiers)*
		Yes, specifically: _____ Direct *(i.e., the participant provided personal details like name and address)* _____ Indirect (*i.e., the participant was given a respondent code to make the participant identifiable)*
Will new data be collected and analyzed along with data from the existing dataset?		Yes **Please answer questions on page 3-5.**
		No

Declaration

We certify that we have read and understand the De La Salle University Code for the Responsible Conduct of Research and will abide by the ethical principles in this document. We will submit a final report of the proposed study to the DLSU-Research Ethics Office. We will not commence with data collection until we receive an ethics review approval from the College Research Ethics Committee.

Name and Signature of Student 1

Name and Signature of Student 2

Name and Signature of Student 3

Name and Signature of Student 4

Endorsement from thesis adviser to the thesis panel for proposal defense…

Name and Signature of Adviser

Date

Endorsement from thesis adviser to the thesis panel for final defense…

This is to certify that the research was conducted in a manner that adheres to ethical research standards. I am thus endorsing the group for final defense.

Name and Signature of Adviser

Date

Appendix H

Informed Consent - Data Gathering

This appendix contains the informed consent document for data gathering.

De La Salle University
College of Computer Studies

You are being asked to participate in a research entitled,

STORYBOOK: GENERATING LIFE STORIES FROM FACEBOOK POSTS

You must be 18 years or older to participate in this data gathering activity. Your participation is voluntary. Please carefully read the information below. Do not hesitate to ask any questions regarding the data gathering process that may not be clear to you.

The data gathering activity is conducted by senior students of B.S. Computer Science, namely Alden Luc Hade, Janica Mae Lam, Camille Alexis Saavedra, and Robee Khyra Mae Te, as part of their requirements for completing their undergraduate thesis. The StoryBook software is developed under the supervision of Ms. Ethel Ong from the College of Computer Studies.

A. INTRODUCTION/PURPOSE

The aim of our software, StoryBook, is to provide you, a Facebook user, with a tool that you can use to produce a short and interesting biography about yourself, or a life story, using some information that you have provided on your Facebook account. To study the types of posts users share in Facebook, we would need sample posts from your Facebook account for analysis.

No data will be extracted from your Messenger account; comments and shares in posts will also be excluded. We would only gather the following information:

- Name
- Gender
- Birthdate
- Educational Background
- Current Location and Hometown
- Preferences (liked pages/artists/musicians)
- Caption/Description of each of your posts
- Tags in each post (if any)
- Number of likes for each post
- Check-ins

B. PROCEDURE

To participate in this data gathering activity, we would like to request that you allow us to perform the following:

1. To scan through your Facebook posts for analysis.
2. To find posts that satisfy the following criteria:
    a. Posts containing what you are doing or how you are feeling (e.g. Travelling to, Feeling, Eating, among others)
    b. Top 5 most liked photos that you have posted (with the assumption that the photo contains a description and/or a caption),
    c. Top 5 most liked posts, and
    d. Top 5 most liked check-ins.
3. To analyze the contents of these posts in order to determine how they can be used to generate a life story.

Your Facebook posts will be used to determine key areas in the design of the software, namely:

1. None of your private information shows up during analysis of the posts (i.e. all that shows up will be what you approved);

2. The criteria for choosing the posts to extract will be enough in generating the contents of the life story; and finally,
3. The generated story is coherent and has good content based on the posts used.

Upon your consent, we will use and store the posts that meet the stated criteria for research purposes only. For confidentiality, we will anonymize the posts by removing any names and replacing these with placeholders.

C. POTENTIAL RISKS AND DISCOMFORT

The data extraction process requires the use of only a personal computer with access to the Internet; therefore, this process will not pose any potential physical risks nor harm to you.

D. POTENTIAL BENEFIT TO SUBJECTS AND/OR TO SOCIETY

Your participation in this activity will provide insights to the project team on the potential use of the posts by Facebook users who wish to generate stories about themselves based on the data that they have posted.

E. CONFIDENTIALITY

Any information that is obtained in connection with this study and that can be identified with you **will remain confidential** and will be disclosed only with your permission, or as required by law. The Facebook posts you posted will be stored and names of people from these posts will be replaced with placeholders.

F. PARTICIPATION AND WITHDRAWAL

Your participation in this study is completely voluntary. If at any point in time you wish to withdraw during the data gathering process, you may do so without penalty or consequence of any kind. Any data collected, should you withdraw, will be disposed of properly.

CONSENT TO PARTICIPATE IN DATA GATHERING FOR "STORYBOOK: GENERATING LIFE STORIES FROM FACEBOOK POSTS"

I confirm that (please tick the appropriate boxes):

	TAKING PART IN THIS PROJECT	
1.	I have read and understood the information about the research as provided in the above Information Sheet.	☐
2.	I have been given the opportunity to ask questions about my participation in the data gathering of my Facebook posts for analysis by the proponents.	☐
3.	I voluntarily agree to participate in the data gathering of my Facebook posts and am aware that taking part in it includes analyzing the contents of my Facebook posts for use in the design of the Storybook software.	☐
4.	I understand that my participation is voluntary; I can withdraw from the data gathering activity at any time.	☐
	USE OF THE INFORMATION I PROVIDE FOR THIS AND FUTURE STUDIES	
5.	The procedures regarding confidentiality have been clearly explained to me.	☐
6.	I agree for the data I provide be archived by the researchers only for research purposes.	☐
7.	Select only **one** of the following:	
	a. I am allowing other researchers to have access to the data collected from this study if they agree to preserve the confidentiality of the data and to the terms I have specified in this form.	☐
	a. I am allowing my name to remain as is during the analysis of the posts extracted from my Facebook account.	☐
	b. I would prefer that the researchers anonymize my Facebook posts and to replace the names with placeholders.	☐
	b. I am **not** allowing other researchers to have access to this data and consent only of its use to this study.	☐

PARTICIPANT:

_____	_____	_____
Name of Participant	Signature	Date

RESEARCHERS:

Alden Luc R. Hade	Janica Mae M. Lam	Camille Alexis T.R. Saavedra	Robee Khyra Mae J. Te
Name and Signature of Researcher	Name and Signature of Researcher	Name and Signature of Researcher	Name and Signature of Researcher

Appendix I

Informed Consent - Testing

This appendix contains the informed consent document for testing.

De La Salle University
College of Computer Studies

You are being asked to participate in a research entitled,

STORYBOOK: GENERATING LIFE STORIES FROM FACEBOOK POSTS

You must be 18 years or older to participate in this end user testing activity. Your participation is voluntary. Please carefully read the information below. Do not hesitate to ask any questions regarding the end user testing that may not be clear to you.

The end user testing activity is conducted by senior students of B.S. Computer Science, namely Alden Luc Hade, Janica Mae Lam, Camille Alexis Saavedra, and Robee Khyra Mae Te, as part of their requirements for completing their undergraduate thesis. The StoryBook software is developed under the supervision of Ms. Ethel Ong from the College of Computer Studies.

A. INTRODUCTION/PURPOSE

The aim of our software, StoryBook, is to provide you, a Facebook user, with a tool that you can use to produce a short and interesting biography about yourself, or a life story, using some information that you have provided on your Facebook account. **The life story is comprised of three main sections - introduction, body and conclusion.** The **introduction** will contain your personal information extracted directly from the "About Me" of your profile such as your name, birthdate, educational background, list of preferences, among others. The **body** will contain the generated story from your life event/s posted in your Facebook account and within the coverage period that you specified. The types of posts include: [1] posts containing what you are doing or how you are feeling (e.g. Travelling to, Feeling, Eating, among others), [2] top 5 most liked photos that you have posted (with the assumption that the photo contains a description and/or a caption), [3] top 5 most liked posts, and [4] top 5 most liked check-ins. Lastly, the **conclusion** ties your story together providing a general idea about you based on the all gathered information.

No data will be extracted from your Messenger account; comments and shares in posts will also be excluded. We would only gather the following information:

- Name
- Gender
- Birthdate
- Educational Background
- Current Location and Hometown
- Preferences (liked pages/artists/musicians)
- Caption/Description of each of your posts
- Tags in each post (if any)
- Number of likes for each post
- Check-ins

B. PROCEDURE

To participate in this testing activity, we would require you to perform the following:

1. You will enter your **login** credentials in Facebook. This part is handled by Facebook's technology; we do **not** have access to your password and email addresses.
2. You will/may **need** to approve the permissions which will allow the software to access your Facebook data, for use in generating your biography. These permissions will allow the system to access your personal information as well as your posts in Facebook.
3. The software will then generate a life story about you, detailing some things about yourself as well as the things you like on Facebook. Please carefully **examine** and **review** the story that is generated.

4. You will have the option to save the generated story for future use in a text file. You will be responsible for keeping the file and/or its dissemination.
5. After using the software to generate your biography, you will answer an evaluation form and participate in a short interview to tell us about your experience and your feedback regarding the resulting story.

Your feedback to us regarding your experience will be used to determine the correctness of the software in key areas, namely:

1. Ensuring that only those information that you made public and had approved for use in this research will show up in the generated story;
2. Validating the appropriateness and correctness (in terms of grammar) of the language used in the generated story;
3. The generated story is coherent and has good content based on the extracted information; and finally,
4. The story is interesting to read.

Note that we will **not** perform any video nor audio recordings while you are using the software and doing the debriefing interview. Upon your consent, the generated story will be stored for use in research purposes only. For confidentiality, we will anonymize the generated story by removing any names and replacing these with placeholders. After storing the generated story, we will show a proof that the anonymized story is indeed the one stored in our system.

C. POTENTIAL RISKS AND DISCOMFORT

The software requires the use of only a personal computer with access to the Internet; therefore, the use of the software will not pose any potential physical risks nor harm to you. Your only difficulties, if any, may arise from usability issues and inability to understand the generated story.

D. POTENTIAL BENEFIT TO SUBJECTS AND/OR TO SOCIETY

Your participation in this activity will provide insights to the project team on the potential use of the software by Facebook users who wish to generate stories about themselves based on the data that they have posted.

E. CONFIDENTIALITY

Any information that is obtained in connection with this study and that can be identified with you **will remain confidential** and will be disclosed only with your permission, or as required by law. The generated story about you will be stored and names of people from the generated life story will be replaced with placeholders.

F. PARTICIPATION AND WITHDRAWAL

Your participation in this study is completely voluntary. If at any point in time you wish to withdraw during testing or during the interview, you may do so without penalty or consequence of any kind. Any data collected, should you withdraw, will be disposed of properly.

CONSENT TO PARTICIPATE IN "STORYBOOK: GENERATING LIFE STORIES FROM FACEBOOK POSTS"

I confirm that (please tick the appropriate boxes):

	TAKING PART IN THIS PROJECT	
1.	I have read and understood the information about the research as provided in the above Information Sheet.	☐
2.	I have been given the opportunity to ask questions about my participation in the testing of the software.	☐
3.	I voluntarily agree to participate in the end user testing of the software and am aware that taking part in it includes being interviewed and my inputs to the software recorded (as part of the software's generated story).	☐
4.	I understand that my taking part is voluntary; I can withdraw from the end user testing of the software at any time.	☐
	USE OF THE INFORMATION I PROVIDE FOR THIS AND FUTURE STUDIES	
5.	The procedures regarding confidentiality have been clearly explained to me.	☐
6.	I agree for the data I provide be archived by the researchers only for research purposes.	☐
7.	Select only **one** of the following:	
	a. I am allowing other researchers to have access to the data collected from this study if they agree to preserve the confidentiality of the data and to the terms I have specified in this form.	☐
	a. I am allowing my name to remain as is in the generated story.	☐
	b. I would prefer that the researchers anonymize the generated story by replacing the names with placeholders.	☐
	b. I am **not** allowing other researchers to have access to this data and consent only of its use to this study.	☐

PARTICIPANT:

_____	_____	_____
Name of Participant	Signature	Date

RESEARCHERS:

Alden Luc R. Hade	Janica Mae M. Lam	Camille Alexis T.R. Saavedra	Robee Khyra Mae J. Te
Name and Signature of Researcher	Name and Signature of Researcher	Name and Signature of Researcher	Name and Signature of Researcher

Appendix J

Content Description of Database Tables

This appendix contains the list of types in some tables.

Direct Knowledge Table

Table J.1: Direct Knowledge Table.

Type	Description
name	Whole name of the Facebook user
birthday	Birthday of the user
hometown	Hometown of the user
location	Current location of the user
email	The email address used to create the Facebook account
gender	Gender of the user
relationship_status	Relationship status of the user

Educational Background Table

Table J.2: Educational Background Table.

Type
Grade School
High School
College

Family Table

Table J.3: Family Table.

Relationship
Mother
Father
Daughter
Son
Sister
Brother
Aunt
Uncle
Niece
Nephew
Cousin (male)
Cousin (female)
Grandmother
Grandfather
Granddaughter
Grandson
Stepsister
Stepbrother
Stepfather
Stepmother
Stepdaughter
Stepson
Sister-in-law
Brother-in-law
Father-in-law
Mother-in-law
Daughter-in-law
Son-in-law
Pet

Likes Table

Table J.4: Likes Table.

ID	Type
1	Actor/Director
2	Aerospace/Defence
3	Airport
4	Album
5	Amateur Sports Team
6	App Page
7	Appliances
8	Artist
9	Arts/Entertainment/Nightlife
10	Attractions/Things to Do
11	Author
12	Baby Goods/Kids Goods
13	Bags/Luggage
14	Bank/Financial Services
15	Bank/Financial Institution
16	Bar
17	Biotechnology
18	Blogger
19	Board Game
20	Book
21	Book Series
22	Bookshop
23	Building Materials
24	Business Person
25	Business Services
26	Camera/Photo
27	Cars
28	Cars and Parts
29	Cause
30	Chef
31	Chemicals
32	Church/Religious Organisation
33	Cinema
34	Clothing
35	Club
36	Coach
37	Comedian

Continued on next page

Table J.4 – continued from previous page

ID	Type
38	Commercial Equipment
39	Community Organisation
40	Community/Government
41	Company
42	Computers
43	Computers/Technology
44	Concert Tour
45	Concert Venue
46	Consulting/Business Services
47	Concert Venue
48	Dancer
49	Designer
50	DIY
51	Doctor
52	Education
53	Entertainer
54	Entrepreneur
55	Electronics
56	Energy/Utility
57	Engineering/Construction
58	Event Planning/Event Services
59	Farming/Agriculture
60	Fictional Character
61	Film
62	Film Character
63	Food/Beverages
64	Food/Groceries
65	Furniture
66	Games/Toys
67	Government Official
68	Health/Beauty
69	Health/Medical/Pharmaceuticals
70	Health/Medical/Pharmacy
71	Home Decor
72	Hospital/Clinic
73	Hotel
74	Household Supplies
75	Industrials

Continued on next page

Table J.4 – continued from previous page

ID	Type
76	Insurance Company
77	Internet/Software
78	Jewellery/Watches
79	Journalist
80	Kitchen/Cooking
81	Landmark
82	Lawyer
83	Legal/Law
84	Library
85	Local Business
86	Magazine
87	Media/News/Publishing
88	Medications
89	Middle School
90	Mining/Materials
91	Movie
92	Museum/Art Gallery
93	Musician/Band
94	Music Award
95	Music Chart
96	Music Video
97	News Personality
98	Non-governmental Organisation (NGO)
99	Non-profit Organisation
100	Office Supplies
101	Organisation
102	Outdoor Gear/Sporting Goods
103	Patio/Garden
104	Performance Art
105	Pet
106	Pet Services
107	Pet Supplies
108	Phone/Tablet
109	Photographer
110	Political Organisation
111	Political Party
112	Politician
113	Preschool

Continued on next page

Table J.4 – continued from previous page

ID	Type
114	Primary School
115	Producer
116	Product/Service
117	Professional Services
118	Property
119	Public Figure
120	Public Places
121	Radio Station
122	Record Label
123	Restaurant/Cafe
124	Retail and Consumer Merchandise
125	School
126	School Sports Team
127	Scientist
128	Shopping/Retail
129	Small Business
130	Song
131	Software
132	Spas/Beauty/Personal Care
133	Sports League
134	Sports Team
135	Sports Venue
136	Sportsperson
137	Sports/Recreation/Activities
138	Teacher
139	Telecommunication
140	Theatrical Play
141	Tools/Equipment
142	Tours/Sightseeing
143	Transport
144	Transport/Freight
145	Travel/Leisure
146	TV Channel
147	TV Network
148	TV Programme
149	TV Show
150	TV/Film Award
151	University

Continued on next page

Table J.4 – continued from previous page

ID	Type
152	Vehicles
153	Video Game
154	Vitamins/Supplements
155	Website
156	Wine/Spirits
157	Writer

Part of Speech Table

Table J.5: Part of Speech Table.

ID	Part of Speech	Part of Speech Value
1	UNKNOWN	Unknown
2	ADJ	Adjective
3	ADP	Adposition
4	ADV	Adverb
5	CONJ	Conjunction
6	DET	Determiner
7	NOUN	Noun
8	NUM	Cardinal number
9	PRON	Pronoun
10	PRT	Particle or other function word
11	PUNCT	Punctuation
12	VERB	Verb
13	X	Other
14	AFFIX	Affix

Relation Table

Table J.6: Relation Table.

ID	Relation	Relation Value
1	UNKNOWN	Unknown
2	ABBREV	Abbreviation modifier
3	ACOMP	Adjectival complement
4	ADVCL	Adverbial clause modifier

Continued on next page

Table J.6 – continued from previous page

ID	Relation	Relation Value
5	ADVMOD	Adverbial modifier
6	AMOD	Adjectival modifier of an NP
7	APPOS	Appositional modifier of an NP
8	ATTR	Attribute dependent of a copular verb
9	AUX	Auxiliary verb
10	AUXPASS	Passive auxiliary
11	CC	Coordinating conjunction
12	CCOMP	Clausal complement of a verb or adjective
13	CONJ	Conjunct
14	CSUBJ	Clausal subject
15	CSUBJPASS	Clausal passive subject
16	DEP	Dependency
17	DET	Determiner
18	DISCOURSE	Discourse
19	DOBJ	Direct object
20	EXPL	Expletive
21	GOESWITH	Goes with
22	IOBJ	Indirect object
23	MARK	Marker
24	MWE	Multi-word expression
25	MWV	Multi-word verbal expression
26	NEG	Negation modifier
27	NN	Noun compound modifier
28	NPADVMOD	Noun phrase used as an adverbial modifier
29	NSUBJ	Nominal subject
30	NSUBJPASS	Passive nominal subject
31	NUM	Numeric modifier of a noun
32	NUMBER	Element of compound number
33	P	Punctuation mark
34	PARATAXIS	Parataxis relation
35	PARTMOD	Participial modifier
36	PCOMP	The complement of a preposition is a clause
37	POBJ	Object of a preposition
38	POSS	Possession modifier

Continued on next page

Table J.6 – continued from previous page

ID	Relation	Relation Value
39	POSTNEG	Postverbal negative particle
40	PRECOMP	Predicate complement
41	PRECONJ	Preconjunt
42	PREDET	Predeterminer
43	PREF	Prefix
44	PREP	Prepositional modifier
45	PRONL	The relationship between a verb and verbal morpheme
46	PRT	Particle
47	PS	Associative or possessive marker
48	QUANTMOD	Quantifier phrase modifier
49	RCMOD	Relative clause modifier
50	RCMODREL	Complementizer in relative clause
51	RDROP	Ellipsis without a preceding predicate
52	REF	Referent
53	REMNANT	Remnant
54	REPARANDUM	reparandum
55	ROOT	Root
56	SNUM	Suffix specifying a unit of number
57	SUFF	Suffix
58	TMOD	Temporal modifier
59	TOPIC	Topic marker
60	VMOD	Clause headed by an infinite form of the verb that modifies a noun
61	VOCATIVE	Vocative
62	XCOMP	Open clausal complement
63	SUFFIX	Name suffix
64	TITLE	Name title
65	ADVPHMOD	Adverbial phrase modifier
66	AUXCAUS	Causative auxiliary
67	AUXVV	Helper auxiliary
68	DTMOD	Prenominal modifier
69	FOREIGN	Foreign words
70	KW	Keyword
71	LIST	List for chains of comparable items
72	NOMC	Nominalized clause
73	NOMCSUBJ	Nominalized clausal subject

Continued on next page

Table J.6 – continued from previous page

ID	Relation	Relation Value
74	NOMCSUBJPASS	Nominalized clausal passive
75	NUMC	Compound of numeric modifier
76	COP	Copula
77	DISLOCATED	Dislocated relation

Entity Category Table

Table J.7: Entity Category Table.

ID	Category	Category Value
1	UNKNOWN	Unknown
2	PERSON	Person
3	LOCATION	Location
4	ORGANIZATION	Organization
5	EVENT	Event
6	WORK_OF_ART	Work of art
7	CONSUMER_GOOD	Consumer goods
8	OTHER	Other types

Appendix K

Production Rules of the Introductory Part of a Life Story

This appendix contains the prodcution rule to be used in the introductory part of the life story.

Table K.1: Production Rules of the Introductory Part of a Life Story.

Variable	**Production Rule**
INTRODUCTION	<FIRST> <SECOND> <THIRD>
FIRST	<name> <OPTIONAL_CLAUSE> <DESCRIPTION_OF_CURRENT_JOB/EDUCATION> | <name> <OPTIONAL_CLAUSE> <BIRTH_CIRCUMSTANCES>
SECOND	"he is from <hometown>, and is now living in <currently_in>" | "she is from <hometown>, and is now living in <currently_in>" | "he hailed from <hometown>, and is now living in <currently_in>" | "she hailed from <hometown>, and is now living in <currently_in>" | <PAST_EDUCATION> <PAST_EDUCATION_TIME> <DESCRIPTION_OF_CURRENT_JOB/EDUCATION> | <PAST_EDUCATION> <PAST_EDUCATION_TIME>

Continued on next page

Table K.1 – continued from previous page

Variable	Production Rule
THIRD	"he <DESCRIPTION_OF_CURRENT_JOB/EDUCATION>" | "she <DESCRIPTION_OF_CURRENT_JOB/EDUCATION>" | "he is the son of <father> and <mother>" | "she is the daughter of <fathe> and <mother>" | "he is the son of <father OR mother>" | "she is the daughter of <father OR mother>" | "he has <sisters.count> sisters, and <brothers.count> brothers" | she has <sisters.count> sisters, and <brothers.count> brothers " | he has <siblings.count> siblings, namely <for loop print names of siblings>" | "she has <siblings.count> siblings, namely <for loop print names of siblings>" | "he is in a relationship with <significant_other>" | she is in a relationship with <significant_other>" | "he has been in a relationship with <significant_other> since <relationship.start_date>" | "she has been in a relationship with <significant_other> since <relationship.start_date> | "he has been married to <significant_other> since <marriage.start_date>" | "she has been married to <significant_other> since <marriage.start_date>
OPTIONAL_ CLAUSE	"as he/she is called by his/her friends" | <BIRTH_CIRCUMSTANCES> | "usually called <nickname> by his/her friends" | ", a <age>-year old <gender>," | null
BIRTH_ CIRCUMSTANCES	"born on <birth_date>" | "was born on <birth_date>" | "born on <birth_date> in <birth_city>" | "was born on <birth_date> in <birth_city>"
DESCRIPTION_OF_ CURRENT_ JOB/EDUCATION	", and is now <studying> at <institution>." | ", and is now <working> at <institution>." | "worked from <job_start_date> to <job_end_date> at <institution>, <DESCRIPTION_OF_ CURRENT_JOB/EDUCATION>"

Continued on next page

Table K.1 – continued from previous page

Variable	Production Rule
PAST_EDUCATION	"<nickname> graduated grade school in <institution>" | "<nickname> finished grade school in <institution>" | "<nickname> graduated high school in <institution>" | "<nickname> finished high school in <institution>" | "<nickname> graduated college in <institution>" | "<nickname> finished college in <institution>" | "<nickname> got his grade school diploma from <institution>" | "<nickname> got her grade school diploma from <institution>" | "<nickname> got his high school diploma from <institution>" "<nickname> got his college diploma from <institution>" | "<nickname> got her high school diploma from <institution>" "<nickname> got her college diploma from <institution>" | "<first name> graduated grade school in <institution>" | "<first name> finished grade school in <institution>" | "<first name> graduated high school in <institution>" | "<first name> finished high school in <institution>" | "<first name> graduated college in <institution>" | "<first name> finished college in <institution>" | "<first name> got his grade school diploma from <institution>" | "<first name> got her grade school diploma from <institution>" | "<first name> got his high school diploma from <institution>" | "<first name> got her high school diploma from <institution>" | "<first name> got his college diploma from <institution>" | "<first name> got her college diploma from <institution>"
PAST_EDUCTAION_TIME	"last <graduation_time>" | "in <graduation_time>" | null

Appendix L

Template for the System to Follow - Introductory Part

This appendix contains the templates for the system to follow. A sample resulting sentence is also shown in the table.

Table L.1: Template for the system to follow - Introductory part. This list is subject to change.

Variable	**Template/s**	**Sample Sentence**
FIRST	1. <name OR nickname> <optional clause> <description of current job / education> 2. <name OR nickname> <optional clause> <birth circumstances>	1. Robee is a college student currently taking up Bachelor of Science in Computer Science at De La Salle University. 2. Camille Saavedra was born on January 29, 1996.
OPTIONAL CLAUSE	1. "as he/she is called by his/her friends" 2. " (<birth circumstances>)" 3. "(usually called <nickname> by his/her friends) " 4. ", a <age>-year old <gender>, " 5. null	1. Alds, as he is called by his friends, [...] 2. Camille was born on January 29. 3. Ken Hosoya, usually called Hapon by his friends, is [...] 4. Alds, a 19-year old boy, [...]

Continued on next page

Table L.1 – continued from previous page

Variable	Template/s	Sample Sentence
BIRTH CIRCUMSTANCES	1. ["born" OR "was born"] + "on <birth_date>" 2. ["born" OR "was born"] + "on <birth_date> in <birth_city>"	1. Camille was born on January 29, 1996. 2. Alds was born on December 4, 1996 in Quezon City, Philippines.
SECOND	1. <he/she> is from <hometown>, and is now living in <currently_in>. 2. <he/she> hailed from <hometown>, and is now living in <currently_in>. 3. <past education> <past education time> <current_education / work> 4. <past education> <past education time>	1. Camille is from Zamboanga City, and is now living in Manila. 2. Airic hailed from Camarines Sur, and is now living in Quezon City. 3. Robee graduated high school in Chiang Kai Shek College in 2013 and is now studying at De La Salle University. 4. Robee graduated high school in Chiang Kai Shek College in 2013.
PAST EDUCATION	1. "<nickname OR first name> <graduated/finished> <high school OR grade school OR college> in ¡institution>" 2. "<nickname OR first name> got <his/her> <high school OR grade school OR college> diploma from <institution>"	1. Alds graduated high school in St. Josephs College of Quezon City. 2. Janica got her college diploma from De La Salle University.
PAST EDUCATION TIME	1. null 2. "last <graduation_time>" 3. "in <graduation_time>"	1. N/A 2. last 2013 3. in 2013

Continued on next page

Table L.1 – continued from previous page

Variable	Template/s	Sample Sentence
DESCRIPTION OF CURRENT JOB/EDUCATION	1. “, and is now <studying / working> at <institution>.” 2. “worked from <job_start_date> to <job_end_date> at <institution>, <current_education / work>.”	1. , and is now studying at DLSU. 2. Alds worked from May to July 2015 at Accenture, and is now studying at De La Salle University-Manila.
THIRD	1. <he/she> <current_education / work>. 2. <he/she> is the <son/daughter> of <father> and <mother>. 3. <he/she> is the <son/daughter> of <father OR mother>. 4. <he/she> has <sisters.count> sisters, and <brothers.count> brothers. 5. <he/she> has <siblings.count> siblings, namely <for loop print names of siblings>. 6. <he/she> is in a relationship with <significant_other>. 7. <he/she> has been in a relationship with <significant_other> since <relationship.start_date>. 8. <he/she> has been married to <significant_other> since <marriage.start_date>.	1. [already mentioned] 2. Alds is the son of Albert and Lilia Geraldine. 3. Alyana is the daughter of Bunnie. 4. Janica has three sisters and two brothers. 5. Janica has five siblings, namely, Angeline, Christly, Kimberly, Jenry, and Jonas. 6. Alds is in a relationship with Alyana Olivar. 7. Yel has been in a relationship with Lance since 2014. 8. Maine has been married to Alden since October 2016.

Appendix M

Content Description of Database Tables

This appendix contains the list of types in some tables.

Keywords Table

Table M.1: Keywords List

Keywords
thirsty
fluid
consume
liquor
alcoholic
quench
thirst
slage
drunk
hydrate
beverage
glass
refreash
liquid
can
liquer
bottle
Continued on next page

Table M.1 – continued from previous page

Keywords
alleviate
water
moisture
take
swallow
cup
mouth
drink
parch
drinkable
beer
lemonade
milk
fill
hydration
taste
soda
round
pour
rehydrate
refreshment
sip
straw
wine
cocktail
coke
fuzz
glasses
guzzle
juice
rum
spritzer
tea
vodka
tonic

Continued on next page

Table M.1 – continued from previous page

Keywords
bartender
cola
mix
bar
coffee
drunken
gulp
intake
ingest
nonalcoholic
orange
apple
pop
refresh
aqua
booze
fizzy
lager
slurp
shake
satiate
inumin
inom
champagne
drinker
imbiber
imbibition
potable
boozer
wassailer
salutation
pledge
salute
toast
toper

Continued on next page

Table M.1 – continued from previous page

Keywords
wassail
drunkenness
boozing
crapulence
deglutition
tope
fuddle
imbibe
special
occasion
happy
party
enjoy
win
year
anniversary
happiness
invite
joy
marry
toast
fun
happen
happening
time
acknowledge
accomplishment
victory
baby
enjoyment
celebrate
congrats
cheer
celebration
parade

Continued on next page

Table M.1 – continued from previous page

Keywords
chirstmas
independence
day
merry
congratulation
congratulations
wild
spring
break
holiday
activity
contest
christmas
eve
valentine
easter
service
new
thanksgiving
greet
friendsversary
commemoration
gift
promotion
birthday
success
bday
b-day
celebrator
observe
observance
fete
eat
chew
hunger

Continued on next page

Table M.1 – continued from previous page

Keywords
consume
food
swallow
taste
hungry
cook
meal
appetite
digest
dinner
lunch
bite
dine
procure
cookie
humburger
burger
dessert
savor
course
chopstick
eaten
ate
consumption
intake
mouth
supper
soup
kain
kainin
kumain
breakfast
snack
eater
feeder

Continued on next page

Table M.1 – continued from previous page

Keywords
feed
depletion
exhaustion
corrosion
rust
deplete
exhaust
corrode
work
go
move
house
passport
place
visit
new
foreign
land
faraway
somewhere
location
historic
site
transportation
mode
jet
lag
historical
sight
get
away
world
see
explore
drive

Continued on next page

Table M.1 – continued from previous page

Keywords
relocation
experience
culture
fly
airplane
reach
destination
taxi
stay
motel
tent
camper
rv
hostel
truck
bus
train
plane
travelling
travel
traveling
detour
trip
sidetrip
roadtrip
road
path
city
area
territory
state
country
way
boat
ride

Continued on next page

Table M.1 – continued from previous page

Keywords
sightsee
book
holiday
ticket
adventure
voyage
venture
itinerary
board
vacation
vacay
abroad
journey
paglalakbay
mover
traveller
traveler
locomotion
motion
movement
locomote
around
journeyer
tripper
jaunt
Thank you
Get
New
Pay
Salary
Buy
Birthday
Celebrate
God bless
Bless

Continued on next page

Table M.1 – continued from previous page

Keywords
Wish
Happy
Season
Watch
Netflix
iflix
TV Series
TV
Movie
Film
Cinema
Theatre
Finished watching
Play
Video game
Game
Platinum
Trophy
Achievement
Finished playing
PS4
XBOX
PC
Vita
Cook
Eat
Dine
Breakfast
Lunch
Dinner
Chicken
Burger
Grill
Bake

Continued on next page

Table M.1 – continued from previous page

Keywords
Fry
Knit
Do
Create
Make
Craft
Cheers
Drink
Drunk
Club
Bar
Beer
Shot
Beer pong
Listen
Music
Sound
Album
Artist
Chart
Read
Book
Finished reading
Done reading
Chapter
Author
Publish
Book series
Go
Travel
At
Visit
Drive
Road
Place

Continued on next page

Table M.1 – continued from previous page

Keywords
Far
Miss
Remember
Reminisce
Here
Attend
Event
Think
Should
Maybe
Probably
Opinion
Honest
Funny
Wow
Haha
:))

References

Abdulkader, A., Lakshmiratan, A., & Zhang, J. (2016). *Introducing deeptext: Facebook's text understanding engine.* Retrieved from https://code.facebook.com/posts/181565595577955/introducing-deeptext-facebook-s-text-understanding-engine/

Adolfo, B. T., Lao, J., Rivera, J. P., Talens, J. Z., & Ong, E. (2015). *Enhancing character interaction in a multiple-character story generation.*

Ang, K., Antonio, J., Sanchez, D., Yu, S., & Ong, E. (2010). Generating stories for a multi-scene input picture. In *Proceedings of the 7th national natural language processing research symposium* (pp. 21–26).

Ang, K., & Ong, E. (2010). *Enhancing event-based semantics in the ontology of picture books 2.*

Boyd, D. M., & Ellison, N. B. (2010). Social network sites: definition, history, and scholarship. *IEEE Engineering Management Review*, *38*(3), 16-31. doi: 10.1109/EMR.2010.5559139

Chaffey, D. (2016). *Global social media statistics summary 2016.* Retrieved from http://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/

Chen, H. W., Lim, M. G., Perez, P. B., Reyes, J. P., & Lim, N. R. (2008). Natural language generation of museum object descriptions based on user model. In *Paclic* (pp. 141–150).

Cheung, C. M., Chiu, P.-Y., & Lee, M. K. (2011). Online social networks: Why do students use facebook? *Computers in Human Behavior*, *27*(4), 1337–1343.

Choudhury, S., & Alani, H. (2014a). Personal life event detection from social media.

Choudhury, S., & Alani, H. (2014b). Personal life event detection from social media. In F. Cena, A. S. da Silva, & C. Trattner (Eds.), *Ht (doctoral consortium / late-breaking results / workshops)* (Vol. 1210). CEUR-WS.org. Retrieved from http://dblp.uni-trier.de/db/conf/ht/ht2014dc.html#ChoudhuryA14

Chua, H., Cu, G., Ibarrientos, C., & Paguilinan, M. (2016). *Alice: Promoting collaborative story writing with a virtual peer.*

Crymble, A. (2010). An analysis of twitter and facebook use by the archival community. *Archivaria*, *70*(2), 125–151.

Darwell, B. (n.d.). *Facebook brings new status updates to mobile so you can share what youre watching, eating, feeling and more on the go.* Retrieved from http://www.adweek.com/socialtimes/facebook-brings-new-status-updates-to-mobile-so-you-can-share-what-youre-watching-eating-feeling-and-more-on-the-go/293460

Doughty, E. (2015). *Facebook is the most popular social network for the over 50s.* Telegraph Media Group. Retrieved from http://www.telegraph.co.uk/goodlife/11751851/facebook-is-the-most-popular-social-network-for-the-over-50s.html

Ellison, N. B., Steinfield, C., & Lampe, C. (2007). The benefits of facebook friends: social capital and college students use of online social network sites. *Journal of Computer-Mediated Communication*, *12*(4), 1143–1168. doi: 10.1111/j.1083-6101.2007.00367.x

Facebook. (n.d.-a). *Javascript sdk - examples.* Retrieved from https://developers.facebook.com/docs/javascript/examples

Facebook. (n.d.-b). *Login dialog.* Retrieved from https://developers.facebook.com/docs/reference/javascript/fb.login/v2.8

Facebook. (2012). *Downloading your info.* Retrieved from https://www.facebook.com/help/131112897028467

Facebook. (2013). *Access tokens - facebook login - documentation - facebook for developers.* Retrieved from https://developers.facebook.com/docs/facebook-login/access-tokens/

Facebook. (2016). *The graph api.* Retrieved from https://developers.facebook.com/docs/graph-api

Farahbakhsh, R., Han, X., Cuevas, ., & Crespi, N. (2013). Analysis of publicly disclosed information in facebook profiles. In *Advances in social networks analysis and mining (asonam), 2013 ieee/acm international conference on* (p. 699-705).

Gervás, P. (2009). Computational approaches to styorytelling and creativity. *AI Magazine. Association for the Advancement of Artificial Intelligence*, 49-62.

Gervás, P. (2012). Story generator algorithms. *The Living Handbook of Narratology. Hamburg: Universidade de Hamburgo. Disponível em: http://www.lhn. uni-hamburg. de/article/story-generator-algorithms Acesso em*, *19*.

Google. (2016). Retrieved from https://cloud.google.com/natural-language/

Gopnik, A. (2012). *Can science explain why we tell stories?* Retrieved from http://www.newyorker.com/books/page-turner/can-science-explain-why-we-tell-stories

Hamilton, F. S. (n.d.). *How to evaluate students writing.* Retrieved from http://creation.com/how-to-evaluate-students-writing

Harden, S. (2016). *Facebook statistics.* Retrieved from http://www.statisticbrain.com/facebook-statistics/

(1999). Retrieved from http://psych.utoronto.ca/users/reingold/courses/ai/cyc.html

Hughes, D. J., Rowe, M., Batey, M., & Lee, A. (2012). A tale of two sites: Twitter vs. facebook and the personality predictors of social media usage. *Computers in Human Behavior*, *28*(2), 561–569.

Indurkhya, N., & Damerau, F. J. (2010). *Handbook of natural language processing* (Vol. 2). CRC Press.

Jain, A., Kasiviswanathan, G., & Huang, R. (2016). Towards accurate event detection in social media: A weakly supervised approach for learning implicit event indicators..

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). *fasttext.* Retrieved from https://research.facebook.com/blog/fasttext/

Keyling, T., & Jnger, J. (2016). *Facepager. an application for generic data retrieval through apis.* Retrieved from https://github.com/strohne/Facepager

Kinsella, S., Passant, A., & Breslin, J. G. (2011). Topic classification in social media using metadata from hyperlinked objects. In *Advances in information retrieval - 33rd european conference on IR research, ECIR 2011, dublin, ireland, april 18-21, 2011. proceedings* (pp. 201–206). Retrieved from https://doi.org/10.1007/978-3-642-20161-5_20 doi: 10.1007/978-3-642-20161-5_20

Laclaustra, I. M., Ledesma, J. L., Méndez, G., & Gervás, P. (2014). Kill the dragon and rescue the princess: Designing a plan-based multi-agent story generator. In *5th international conference on computational creativity.*

Liu, H., & Singh, P. (2004). Conceptneta practical commonsense reasoning toolkit. *BT technology journal*, *22*(4), 211–226.

Mannes, J. (2016). *Facebook's artificial intelligence research lab releases open source fasttext on github.* Retrieved from https://techcrunch.com/2016/08/18/facebooks-artificial-intelligence-research-lab-releases-open-source-fasttext-on-github/?ncid=rss\&utm_source=feedburner\&utm_medium=feed\&utm_campaign=Feed\%3A\%20Techcrunch\%20(TechCrunch)\&utm_content=FaceBook\&sr_share=facebook

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Inc, P., Bethard, S. J., & Mcclosky, D. (2014). The stanford corenlp natural language processing toolkit. In *In proceedings of the 52nd annual meeting of the association for computational linguistics: System demonstrations* (pp. 55–60).

Mars, N. J. (1995). *Towards very large knowledge bases: Knowledge building & knowledge sharing 1995.* Ios Press.

Martin, J. H., & Jurafsky, D. (2000). Speech and language processing. *Interna-*

tional Edition, *710*.

Mawhorter, P. (2013). Reader-model-based story generation. In *Ninth artificial intelligence and interactive digital entertainment conference.*

McIntyre, N., & Lapata, M. (2009). Learning to tell tales: A data-driven approach to story generation. In *Proceedings of the joint conference of the 47th annual meeting of the acl and the 4th international joint conference on natural language processing of the afnlp: Volume 1-volume 1* (pp. 217–225).

Meehan, J. R. (1977). Tale-spin, an interactive program that writes stories. In *International joint conference on artificial intelligence* (Vol. 77, pp. 91–98).

Méndez, G., Gervás, P., & León, C. (2014). A model of character affinity for agent-based story generation. In *9th international conference on knowledge, information and creativity support systems, limassol, cyprus* (Vol. 11, p. 2014).

Nadkarni, A., & Hofmann, S. G. (2012). Why do people use facebook? *Personality and Individual Differences*, *52*(3), 243 - 249. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0191886911005149`

Overview of cyc inferencing. (1994). Retrieved from `http://www.cyc.com/overview-cyc-inferencing/`

PearAnalytics. (2009). *Twitter study - august 2009.* Retrieved from `https://web.archive.org/web/20110715062407/www.pearanalytics.com/blog/wp-content/uploads/2010/05/Twitter-Study-August-2009.pdf`

Pehcevski, J., & Piwowarski, B. (2009, May). Evaluation metrics for structured text retrieval. In M. T. Özsu & L. Liu (Eds.), *Encyclopedia of database systems* (pp. 1015–1024). Springer.

Pradhan, K. (2010). Retrieved from `https://www.facebook.com/notes/terabug/download-all-facebook-photos-status-wall-posts-together-in-zip-file/10150118571353989/`

Rao, L. (2010). *Twitter added 30 million users in the past two months.* Retrieved from `https://techcrunch.com/2010/10/31/twitter-users/`

Reiter, E., & Dale, R. (1997). Building applied natural language generation systems. *Natural Language Engineering*, *3*(01), 57–87.

Setty, S., Jadi, R., Shaikh, S., Mattikalli, C., & Mudenagudi, U. (2014). Classification of facebook news feeds and sentiment analysis. In *Advances in computing, communications and informatics (icacci, 2014 international conference on* (pp. 18–23).

Sleimi, A., & Gardent, C. (2016). Generating paraphrases from dbpedia using deep learning. *WebNLG 2016*, 54.

Solis, C. J., Siy, J. T., Tabirao, E., & Ong, E. (2009). Planning author and character goals for story generation. In *Proceedings of the workshop on computational approaches to linguistic creativity* (pp. 63–70).

Speer, R., & Havasi, C. (2012). Representing general relational knowledge in conceptnet 5. In *Lrec* (pp. 3679–3686).

Staff, M.-H. (2000). *Writer's choice writing assessment and evaluation rubrics*

grade 9. McGraw-Hill Education. Retrieved from https://books.google.com.ph/books?id=v5mNpk49FMkC

Syliongka, L. R., Oco, N., Lam, A. J., Soriano, C. R., Roldan, M. D. G., Magno, F., & Cheng, C. (2015). Combining automatic and manual approaches: Towards a framework for discovering themes in disaster-related tweets. In *Proceedings of the 24th international conference on world wide web* (pp. 1239–1244).

Technopedia. (2016). Retrieved from https://www.techopedia.com/definition/25328/data-extraction

Tuffield, M. M., Millard, D. E., & Shadbolt, N. R. (2006). Ontological approaches to modelling narratives. In *2nd akt dta symposium, akt. aberdeen.* University, UK.

Varol, O., Ferrara, E., Davis, C. A., Menczer, F., & Flammini, A. (2017). Online human-bot interactions: Detection, estimation, and characterization. *CoRR*, *abs/1703.03107*. Retrieved from http://arxiv.org/abs/1703.03107

Wang, H., Can, D., Kazemzadeh, A., Bar, F., & Narayanan, S. (2012). A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the acl 2012 system demonstrations* (pp. 115–120).

Weaver, J., & Tarjan, P. (2013). Facebook linked data via the graph api. *Semantic Web*, *4*(3), 245–250.

West, L. E. (2013). Facebook sharing: A sociolinguistic analysis of computer-mediated storytelling. *Discourse, Context and Media*, *2*(1), 1-13. Retrieved from http://www.sciencedirect.com/science/article/pii/S2211695812000682

Westlake, E. (2008). Friend me if you facebook: Generation y and performative surveillance. *TDR/The Drama Review*, *52*(4), 21–40.

Widrich, L. (2012). *The science of storytelling: Why telling a story is the most powerful way to activate our brains.* Retrieved from http://lifehacker.com/5965703/the-science-of-storytelling-why-telling-a-story-is-the-most-powerful-way-to-activate-our-brains

Youse. (2005). *Ten elements of biography.*

Zhang, X., & LeCun, Y. (2015). Text understanding from scratch. *arXiv preprint arXiv:1502.01710*.