USER-DRIVEN STORY GENERATOR USING PUBLIC KNOWLEDGE BASE

TAY KAE YI

MASTER OF INFORMATION SYSTEMS

LEE KONG CHIAN FACULTY OF ENGINEERING AND SCIENCE

UNIVERSITY TUNKU ABDUL RAHMAN

SEPTEMBER 2015

USER-DRIVEN STORY GENERATOR USING PUBLIC KNOWLEDGE BASE

TAY KAE YI

A project report submitted in partial fulfillment of the

requirements for the award of

Master of Information Systems

Lee Kong Chian Faculty of Engineering and Science

Universiti Tunku Abdul Rahman

September 2015

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _____

Name : _____

ID No. : _____

Date : _____

APPROVAL FOR SUBMISSION

I certify that this project report entitled "**USER-DRIVEN STORY GENERATOR USING PUBLIC KNOWLEDGE BASE**" was prepared by **TAY KAE YI** and has met the required standard for submission in partial fulfillment of the requirements for the award of Masters of Information Systems at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor : _____

Date : _____

ABSTRACT

Storytelling is a key aspect of movies or games, as an interesting story can easily bring more depth and capture a larger audience. Stories can also be used as a form of education for children. Hence there is a need to continuously generate new and more creative stories. In order to fulfill these demands, the industries look towards the information technology where computers are introduced as story generators. A "story generator" is a software application or program which uses a certain algorithm or approach to generate stories of a given domain based on selected input parameters. With the advancement of artificial intelligence and computational power, various story generation algorithms were introduced and made possible. Some of the existing story generation algorithms include data-driven algorithm, case-based algorithm and role-based algorithm. Each approach has its own limitations such as repetitive stories, lack of plot progression and lack of user interaction. The main objective of this research is to develop a user-driven approach for story generation which allows more user interaction. This approach promotes user interaction by probing questions to users and the responses given by the users are used determine the story progression. The story generator developed in this research is based on existing public knowledge bases. Existing public knowledge bases are used in order to cut down on development time. The outcome of this research is a simple prototype application which implements the user-driven approach for story generation. A survey is done in order to evaluate the level of user interaction of the user-driven story generator as compared to existing story generators. The evaluation results show that the user-driven story generator has a higher level of user interactions which is important in maintaining the user's interest towards the generated stories.

CONTENTS

CHAPTER 1

INTRODUCTION

Story telling is an important component in the world of games. In most genres, a game's storyline is what captures the attention of gamers and it becomes the selling point of the game (Gervas, Diaz-Agudo, Peinado & Hervas, 2005). Also, storytelling can play a huge role in a child's growth where it can be used as a form of education for the child (McIntyre & Lapata, 2009). Making use of technologies, story generation can be made automated.

With the advancement of technologies and artificial intelligence, there are more and more interactive-based and automatic story generators being developed. There is already usage of automated tools for plot development in the entertainment industry. Besides that, dynamic story generation can be used as a form interactive learning in classroom teachings or online-based educational sites. Children can engage and interact with the program in order to generate stories dynamically (McIntyre & Lapata, 2009). This may help to attract a child's interest and promote creativity. There are various types of story generators available today, each with different story generation techniques and algorithms.

Some of the existing story generation algorithms, such as the case-based story generator, make use of huge sets of previous stories and come up with new stories using mix-and-match method. There are also story generators which use the data-driven story generation algorithm. This algorithm takes in the given user-inputs and used it to find co-

related verbs, nouns, adjectives and relations from a huge knowledge base (McIntyre & Lapata, 2009). Sentences using the retrieved words are formed based on the level of co-relation and these sentences are then joined together to form a story. These story generation algorithms and approaches are further discussed in Section 2. One of the limitations of these existing story generators is the lack of user interaction, which is important to maintain a user's interest towards the generated story. Finding a new story generation technique which promotes better user interaction and also to develop the story generator within a short time frame are some of the key challenges of this research.

Most of the current story generators use a knowledge base, either in the form of a huge database or ontology, which is *trained* with words, relationships, nouns and verbs. Training a knowledge base consumes a huge amount of effort and time (McIntyre & Lapata, 2009). Also, in order to keep the knowledge base up-to-date, more time has to be invested to continuously upgrade and enhance it. In this research, existing public knowledge base is used. This is to help reduce the development time of the knowledge base. More information of the selected public knowledge base can be found in Section 4.3.

1.1 Problem statement

There are various types of existing story generators, each can be categorized based on the story generation algorithm it uses. Some of the algorithms include the case-based algorithm, role-base algorithm and data-driven algorithm. Each of these algorithms has its own benefits and limitations.

One of the key problems of existing story generation algorithms is the limitation of user interaction. User interaction is important in maintaining a user's interest towards the story, especially in games for children (Solis et al, 2009). For example, the data-driven story generator generates sentences based on highest score between words'

correlation (refer Section 2.2.1). However, the generator does not have fluency and coherence between sentences. This makes the story progression feels inconsistent and does not have fluency (McIntyre & Lapata, 2010). The user does not have control or influence over the story progression, as the story generator does not take in user input during the story generation process. This is also the case for other story generators such as case-based story generator or role-based story generator.

Another limitation which can be seen in existing story generators is the knowledge base of the story generator. The knowledge base is a key component of the story generators, and it stores information on characters, story plots, adjectives, verbs, nouns and story settings (McIntyre & Lapata, 2009). This means the number of story variations generated depends on the story generator's knowledge base. A story generator with small knowledge base will only be able to generate limited amount of stories. A lot of development time is spent in creating the knowledge base of a story generator (McIntyre & Lapata, 2009). Even after the creation of the knowledge base, the developers have to continuously maintain and enhance the base.

The motivation behind this research is to come up with a story generation approach which allows more user interaction and also to make use of existing public knowledge bases in order to reduce development time.

As discussed above, the problems in existing story generators approaches can be summarized as below:

a) The need to provide more user interaction in the story generation process.

b) The lack of fluency and coherence between sentences in a generated story.

c) The huge amount of time and effort needed to create, maintain and enhance the knowledge base of a story generator.

1.2 Research questions

The goal of this research is to develop a user-driven story generation approach which allows more user interaction. The final output is a story generator application which uses existing public knowledge bases and implements the user-driven approach. The general question of this research is:

How to formulate a new user-driven story generation approach and implement it in the story generator?

The sub-questions can be divided as follows:

a) **What are the existing algorithms used for story generation and what are the limitations of these algorithms?**

b) **How to allow more user interaction in the story generation process so that the user's attention is maintained?**

c) **How to design and develop a story generator which implements the user-driven story generation approach?**

d) **How to validate the story generator?**

1.3 Objectives

With the research questions set as above, the focus and objectives of this research are as listed below:

a) **To analyze existing story generation algorithms**

Existing story generation algorithms such as case-based algorithm, data-driven algorithm and role-based algorithm are studied and compared based on several criteria such as levels of user interaction, story creativity, story plot progression, and type of knowledge base used.

b) To formulate and design a new user-driven story generation approach

This is the main objective of the research. In this research an approach which will allow more user interactions will be developed. This approach's aim is to promote more user interaction during the story generation process. In this user-driven story generation approach, the user will have to provide several simple input parameters to the story generator during initialization stage. Based on these inputs, the story generator will determine the story's settings, characters and then generate the story's introduction. The next stage of the approach is to prompt questions to the user and this allows the user to decide or influence how the story progresses. This process will be repeated until the end of story is reached. In short, this will be the basis of the user-drive story generation approach. This approach makes use of existing public knowledge base in order to shorten the development time as compared to creating a new knowledge base from scratch. Another advantage of using public ontology is that the ontology is maintained by the public and it is ever-growing (more information added from time to time). The knowledge bases are used to provide information during the sentence generation process.

c) To develop and implement the user-driven story generation approach into the story generator

The user-driven story generation approach is implemented into a simple story generator application. Also, existing public knowledge bases are analyzed and selected to be used in the story generator. There are multiple noteworthy public knowledge bases that can be considered for use in story generation, such as the Suggested Upper Merged Ontology (SUMO), OpenCyc, WordNet or DOLCE. In this research, the WordNet ontology is chosen as the base ontology. This ontology helps provide verbs, nouns, tenses, adjectives and synonyms. All these are used in

the sentence rendering process. Another public ontology that is used is the "Animal ontology". This ontology provides general information on animals such as their habitat, food, body parts and types of movements.

d) To evaluate the user-driven story generator

Evaluation of the user-driven story generator is done in the form of a survey. As mentioned previously, the key purpose of this research is to develop a new story generation approach which allows more user-interaction. The survey will focus on this aspect. The survey will gather users' ratings and feedbacks on the user-driven story generator as compared to existing story generators such as case-based story generator and data-driven story generator. Among some of the aspects that will be recorded in the survey are the level of user-interaction, the user-experience as well as the quality and variation of the generated stories.

1.4 Scope

The scope for this research will be defined in this section. The few areas in which the scope is limited are:

a) Storyline background and domain

The background and domain of the story generator are limited to *children's fairy tales*. In this research, existing story generators which also focus on simple storylines, preferably on children's fairy tale, are analyzed. This allows easier comparison and evaluation of the story generators. Limiting the domain and background will also help to cut-short development time. The focus of this research is not on the width and broadness of domains that the story generator can cover. Instead the focus of this research is to determine the feasibility of using a new story generation process which is based on user-driven approach.

b) Story plot

The research is kept simple by *limiting the plot to a fixed structure* of problem, rising action, solution and climax. In most existing story generators, the story plot

is fixed. For easier comparison, the developed story generator will also use a fixed plot for the stories generated.

c) Knowledge base

As mentioned in previous section, the knowledge base used for this research will be based on **existing public knowledge bases**. By using existing public knowledge bases, there is no need to spend time and effort on creating an ontology from scratch. The WordNet ontology and the "Animal" ontology are the two public knowledge bases that will be used in this research.

d) Algorithm / approach

The approach developed in this research will be a user-driven approach. This approach will continuously prompt questions to the user during the story generation process. The input from the user will influence how the story progresses.

1.5 Significance

This research can assist in the development of a dynamic story generator. Dynamic story generators will generate stories on the fly and the story varies according to the user's inputs. With the user-driven approach, the story generator will be able to encourage more user-interaction. This will be helpful when applied in educational games for children (Solis et al, 2009) as the children will be more engaged with the stories. Every input from them will change the story's progression, and this helps to stimulate their thinking process.

Also, the user-drive story generator can be used for movie script writing. The story generator is similar to the case-based story generator where it uses a fixed story plot and replaces the key characters, events, actions, adjectives and verbs of the story. This will help generate random new stories and can be used as a tool to help provide ideas for script writing (Gervas, Diaz-Agudo, Peinado & Hervas, 2005).

The story generator can also be used in schools to encourage children to read more stories. Instead of reading stories from books, children can now generate their own stories for reading. This can also be helpful for the "Program NILAM" which was implemented by the Ministry of Education Malaysia. The program's main objective is to encourage book-reading amongst children. By introducing the user-driven story generators, children can be motivated to generate their own stories and indirectly, this will push the children closer towards the goals set by Program NILAM.

CHAPTER 2

LITERATURE REVIEW

In this section, the literature review on existing story generators and their algorithms are studied and analyzed. Analyzing and comparing these story generators will give an overview of each algorithm's strengths and weaknesses. This section will also cover on topic such as ontology and briefly dive into public ontology as it will be used as the knowledge base of the story generator developed from this research.

2.1 Story generators

As mentioned previously, stories can play a large role in games, children's education or even in entertainment industries. This section discusses about existing story generators and the techniques they used in automating story generation.

Most, if not all, story generators make use of a large database or ***ontology*** as their "base knowledge". This *base* will normally contain words categorized into nouns, verbs and adjectives which can be used for sentence generation. A general architecture of a story generator can be seen in the following diagram:

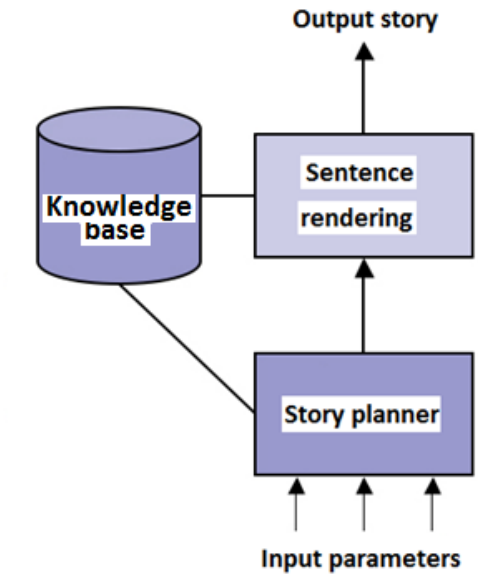


Figure 2.1: General architecture of a story generator

The main modules of a story generator are the "**Knowledge base**", "**Story planner**" and the "**Sentence renderer**".

2.1.1 Story planner

This component is the *engine* of the story generator and where the main algorithm lies. The data retrieval, data manipulation, story-tree generation, and other logic of the story generation resides in this engine. A sub-model of the story planner would be the interface between the engine and the database structure. Depending on the selected database model, different type of interface is used to allow communication between the database and the story generator.

2.1.2 Knowledge base

The knowledge base is the backbone or data provider of the story generator. The base will provide information such as nouns, character traits, actions, adjectives and much more. In order to allow more variation in the generated stories, the knowledge base has to contain huge set of data. Knowledge base can exist in the form of standard database or ontology.

2.1.2.1 Database

Among the types of knowledge base used for story generators include database such as Oracle, Microsoft SQL database and MySQL database. When using the database, standard database development methods such as object-oriented approach can be used. Huge amount of time has to be spent to *inject* information into the database.

2.1.2.2 Ontology

In some story generators, the knowledge base exists in the form of an ontology (Refer Section 2.2.4). Ontology provides a better representation of *relationships* between different entities.

Ontology is a term originally coming from philosophy. It means the philosophical study of the nature of being, existence, or reality, as well as the basic categories of their relations. In short, ontology is a theory of existence.

However, in information systems, the term ontology refers to an explicit specification of conceptualization, as termed by Tom Gruber. Ontology is used to capture information on a certain domain such as structure, relations, restrictions, entities, actions and others (Obitko, "Ontologies and Semantic Web"). In another explanation, ontology is a body of knowledge describing a particular domain. In any particular domain of our world, can be categorized into its objects, concepts, and how they are related to one another. All these are captured in ontology and represented in a formal language. As such, it can be generally said that ontology describes and specifies a particular domain.

2.1.3 Sentence renderer

The sentence renderer is the final main component of a story generator. The sentence renderer helps to construct a proper sentence before the final story can be presented to the user. In this component, emphasis is placed on proper

grammatical usage, fluency of sentence, proper usage of nouns, verbs and adjectives.

The following sections will discuss on different types of databases or ontology used in existing story generator. Analysis is done on the algorithms used and comparison is made between the existing story generators.

2.2 Existing story generators

Numerous automated story generation algorithms exist out there today. Different story generators make use of different story generation algorithms, and each has its own advantages and disadvantages. Depending on the purpose of the story generator, certain algorithm will be better suited. For example, story generator for the purpose of developing prototypes for movie script can make use of the case-based story generation algorithm (Refer 2.2.3). In this section, the focus will be to analyze a few of the story generation techniques and which will be used for comparison with the developed user-driven story generation approach later on.

2.2.1 Data-driven story generator with evolutionary search

A story generator based on data-driven approach, enhanced with plot induction and evolutionary search was introduced by McIntyre and Lapata. This story generator uses a training corpus[1] as its basis for sentence generation (McIntyre & Lapata, 2009, 2010). The corpus was *trained* using a large set of fairy tale stories from Andrew Lang. Word tokens and how the words are related to one another were measured and scored. This knowledge base then becomes the reference point in sentence generation, where it helps to score each sentence based on the *relation of* the words / entities in the sentence. This method ensures correctness of a sentence, but **does not provide fluency or coherence between**

consecutive sentences. Hence to increase the coherence between sentences, several methods were employed, such as extending the training corpus to extract chain events so that it can expect or predict the next chain of events. Also, interest model and coherence model were introduced to calculate the score of the generated stories. Multiple story trees are generated by the story generator, and then the path that gives the highest score were selected as the output story.

[1]The training corpus is a large database "trained" using 437 fairy tale stories from Andrew Lang. It consists of 15,789 word tokens. These tokens are linked and scored based on how one word appears more frequently after another.

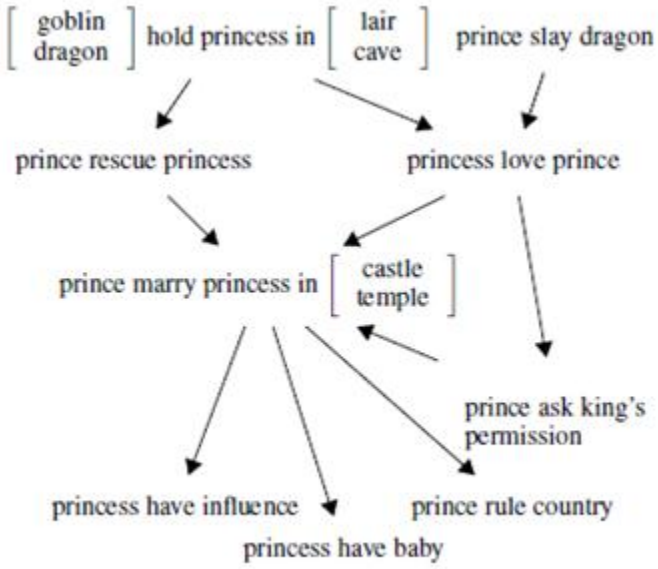


Figure 2.2: Sample plot graph for given input "Princess loves the prince"

Later on, evolutionary search is introduced to further refine the story generator. The generated story trees' nodes are swapped (cross-over) and changed (mutation) resulting in new storylines which may have higher scores (McIntyre & Lapata, 2010). This process is recursive and can be repeated based on the defined rate of cross-over and mutation.

The story generator introduced here has a great algorithm in story generation; however the corpus used is **considered small and constricted**. It was limited to the domain of fairy tales and was based only on 437 stories. Replacing the training corpus with a larger database or ontology may help. Also, the generated sentences and stories can be **considered flat** as plot progression and character expressions were not taken into account. Lastly, the story generator only takes in a few parameters from the user in order to generate the stories. It has **very little user-interaction** and stories generated are very random.

2.2.2 Role-based story generator

RoleModel is a role-based story generator for novels, where it differs from other story generators which can be goal-based, character-based or author-based. It explicitly models roles to generate meaningful variations of story situations (Chen, Smith, Jhala, Wardip-Fruin & Mateas, 2004). This system takes in an input on the timeline, character's traits and names, as well as forbidden or required actions. With an ontology which consists of character traits and attributes, roles, actions and information on actions and reactions, this story generator will be able to generate a story which is to the user's requirements (i.e. plot and flow of actions). However, this shows that **much manual work is needed** to construct and develop the ontology.

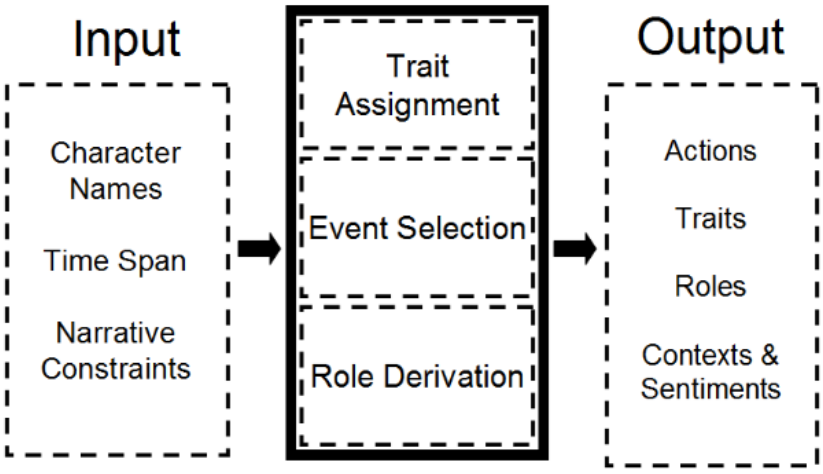


Figure 2.3: RoleModel system architecture

Also interesting to note here is the ability of this generator to generate stories based on either highly constrained requirements or few to no requirements. Different perspectives were also introduced (i.e. victim, aggressor and bystander) in order to provide variation and diversity in the stories. As generated stories are **limited to what the ontology can provide,** a simple step in enhancing this generator will be to increase the data in the ontology. In order to do that, a larger set of actions, characters and contexts have to be added to the ontology. RoleModel is also **restricted to the domain** defined in input hence limiting the variations in the output stories. Cross-domain stories appear as an interesting enhancement for this system. Also, the story generator **does not allow users to change or influence the story generation process**. The stories generated are only dependent on the input parameters from the user in the initial stage.

2.2.3 Case-based story generator

As described above, there can be different story generators depending on whether it is goal-based, role-based, character-based or others. Another interesting research to look at is the story plot generator which has story plots created by a case-based reasoning (Gervas, Diaz-Agudo, Peinado & Hervas, 2005). This story generator, called ProtoPropp, reuses a base of existing stories to generate and produce new stories. It has its own ontology which consists of concepts such as

"Propp functions", "Moves", characters and its properties, roles, places, objects, cases and etc. Existing stories are broken down into components befitting the ontology so that they can be stored and classified. The diagram below shows an example of a story being broken down into parts mentioned above.

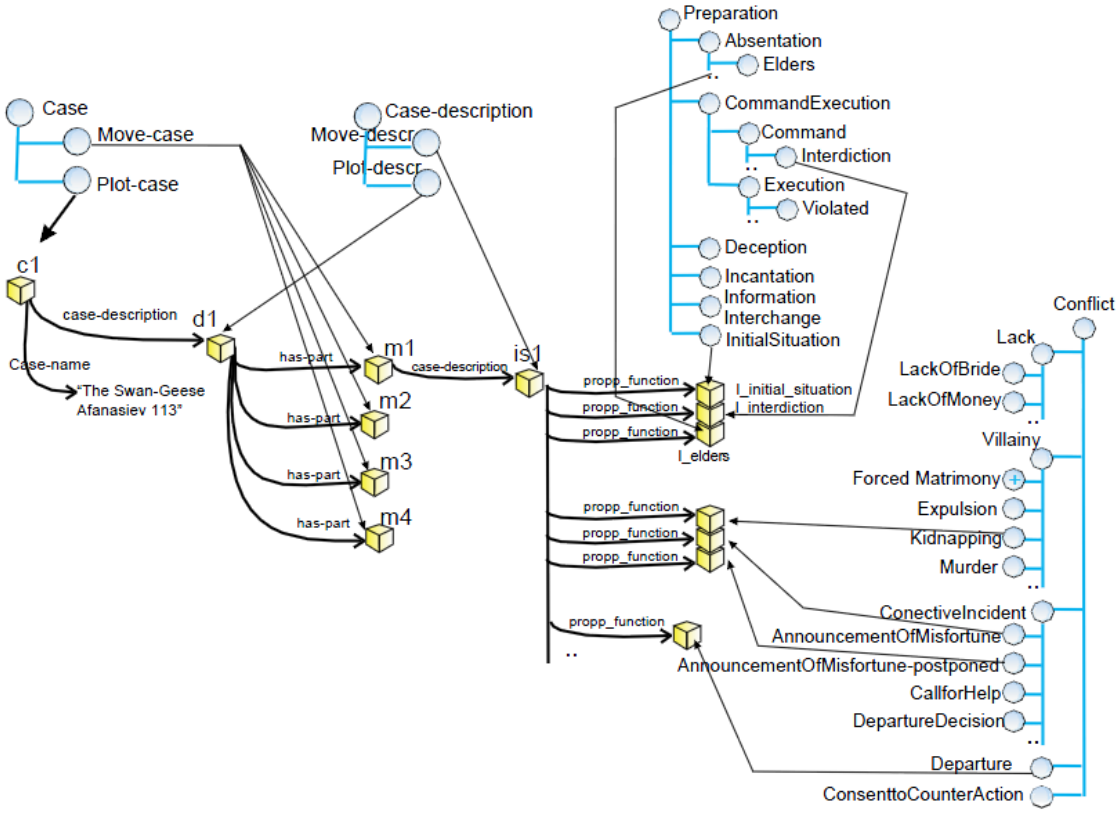


Figure 2.4: Example of a story's components

The creation of this ontology **requires much time and manual work**. However, it can be easily extended to incorporate more data, personas, roles and other information. The story generator firstly considers the input parameters such as the required characters, roles or places. Then it selects an existing "case" from its ontology and replaces different nodes in this case with related parameters generated from the given input. This gives rise to a new story with a nice story plot, flow of events and character development. Although this story generator is **not fully-automated,** as it has **a rigid plot structure**, it demonstrates its purpose as a fast-prototyping tool that could provide a large number of acceptable plots. This can be helpful in the entertainment industry.

2.2.4 Story generator using public ontology

There are several public ontologies currently available. Among them are Suggested Upper Merged Ontology (SUMO), OpenCyc or DOLCE. There are existing story generators such as the Suggested Upper Merged Ontology story generator (SUMOs) which makes use of the public ontology to generate stories. This section discusses the story generator based on Suggested Upper Merged Ontology.

2.2.4.1 SUMOs

Most story generators require a base of "knowledge" where it can extract information for sentence construction or plot generation. The required information can be stored and conceptualized in ontology (Jaya & Uma, 2010). Having a large set of ontology will help to provide more cases and variation in the generated stories. Ontology can be easily enhanced and extended for different plots, environments, characters and much more. The Suggested Upper Merged Ontology (SUMO) is claimed to be the largest formal public ontology in existence today (Pease, 2013). Works have been done to create story generators based on SUMO, an example of which is SUMOs (SUMO stories). SUMOs is an automatic story generator which uses first-order logic (Cua, Ong, Manurung & Pease, 2010) to declaratively describe models of the world. Among advantages of using SUMO include the comprehensive coverage of 20,000 terms and 70,000 axioms. Also, terms in SUMO have been mapped to WordNet.

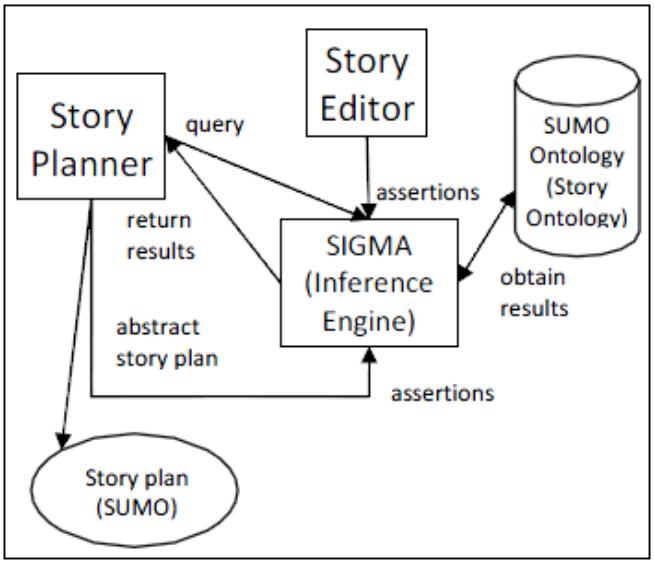


Figure 2.5: Architecture of SUMO story generator

SUMOs uses a plot-based generation introduced by Machado (Machado, 2003), in which a story will always have a problem, rising action, solution and climax. However, in this research, generated stories have **simple sentences** where each of these sentences will always consist of an actor and an action. This leads to a flat tone in the story and has lower coherence between sentences. Besides that, to improve the story generator, more **interesting plots** and **character development** can also be introduced into the system.

2.3 Comparison

In this section, a summary is made for the types of knowledge base used, algorithms, level of user-interaction, performance and quality of generated stories of some of the story generators mentioned above. The selected story generators are the SUMOs (SUMO based story generator), case-driven story generator and the data-driven story generator. These generators use different knowledge base, algorithms and story plot pattern. In the evaluation of the user-driven story generator, comparison is made between the user-driven story generator with the data-driven story generator and case-based story generator. This is important to evaluate and judge how the developed user-driven story generator matches up to existing story generators such as the data-driven story generator and case-based story generator.

2.3.1 Strengths and weaknesses of existing approaches

Highlighted in the literature review, each of these story generators has its own strengths and weaknesses. The table below shows the generators' strengths and weaknesses:

Story generators	**Strengths**	**Weaknesses**
SUMOs	i) Uses public ontology, less development time ii) Easier to scale the story generator to cover other domains	i) Uses a fixed plot. Repetitiveness may cause users to be bored ii) Less emphasis on sentence generation. Generated sentences may have grammatical errors.

Data-driven story generator	i) High coherence between sentences after introducing new function to iterate through different "story tree paths" ii) Large set of vocabularies, nouns and adjectives which are geared towards fairy tale domain, hence very suitable for children's story generation	i) Limited to fairy tale domain ii) Hard to enhance and grow its knowledge base
Case-based story generator	i) Can generate comples story based on existing "story plots" ii) Very smooth story progression and character development. Can be used for script idea generator.	i) Does not render completely new stories. Story plots may repeat. ii) Not fully automated

2.3.2 Summary of existing story generators

The table below helps to categorize existing story generators based on the certain criteria:

	Story generators		
	SUMOs	Data-driven generator	Case-based generator
Knowledge base	Public ontology SUMO	Self-developed. Trained and grown using 437 fairy-tale stories	Self-developed ontology. Developed by breaking down existing stories into desired components.
Approach / algorithm	Based on initial user input to generate sentences	Score and rank method	Case-based reasoning. New stories are creating by replacing entities into an existing set of stories
Story plot pattern	Problem, rising action, climax and solution	Flexible and dynamic	Depending on the selected story

Initial user input	Entities and domain	Topic and length of story	Characters, roles, places
User-interaction	Allows user to select input parameters by selecting pictures in the software.	Only takes in input parameter during initialization stage, before story generation process begins.	Only takes in input parameter during initialization stage, before story generation process begins.

As seen in this section, there are various types of story generators and each has its own strengths and weaknesses. There is no perfect story generator as it is built to fit the purpose of its usage, whether for generating children fairy tales, or for prototyping a movie script, or even for gaming.

Also, the focus of this research is to create a user-driven story generator for child education. For this purpose, it is important to attract children's attention via more user interaction. From the comparison, it can be said that the existing story generators such as the data-driven story generator and case-based story generator lacks in user-interaction. Hence the purpose of the user-driven story generator is to gain an edge over the above-mentioned story generators by providing more user interactions.

In this research, existing public knowledge bases will be used and incorporated into the story generator. Most existing story generators use self-trained knowledge bases, which consumes a lot of effort and time. There is already a story generator (the SUMOs) which uses a public ontology as its knowledge base. However, the SUMOs lack user-interaction and have a different set of story generation algorithm. In this research, the final deliverable will be a user-driven story generator which is based on existing public knowledge bases.

CHAPTER 3

METHODOLOGY

A methodology helps to provide direction, saves time and helps to focus on important aspects of a research. There are various types of methodologies available and choosing a suitable methodology for this research is important. The following sections describe the chosen methodology for this research.

3.1 Engineering Research Methodology

For this research, the *engineering research methodology* (Marcos M.L., 2005) will be used. This methodology consists of the following stages:

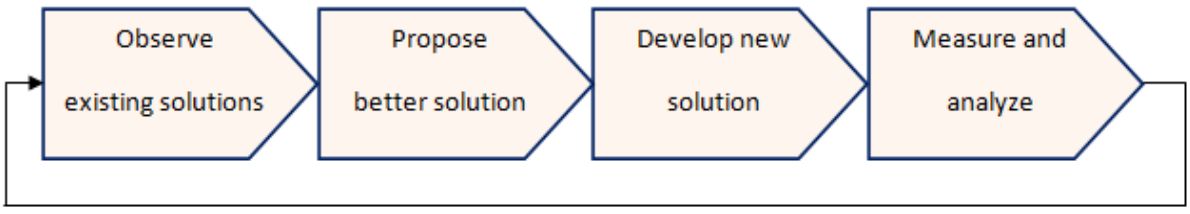


Figure 3.1 Engineering research methodology

In the proposed engineering method, the steps are to firstly observe and analyze existing solutions, then a better solution is proposed, next the solution is developed before measuring and evaluating the solution. These steps are repeated until no improvements are possible.

In this research, the same approach is followed, but without the repetitive iterations from the last stage to the first stage. Each stage of the methodology is described below:

a. Observe existing solutions

i. There are various story generators today, each having its own pros and cons. These story generators are used for different purposes. In order to study these generators, a literature review has been done. In previous sections, the details of some existing story generators are listed down and compared based on criteria as described in Section 2.3. Some of the criteria which can be used for comparison include level of user interaction, type of knowledge base used, sentence coherence, story plot and story progression. These criteria are chosen because they can be evaluated, scored and compared between stories generated by different generators. Also, the objective of the research is to develop a story generator which allows more user interaction. The criteria mentioned above will be used in the survey to evaluate and compare the developed user-driven story generator with existing story generators such as data-driven story generator and case-based story generator.

b. Propose a better solution

i. In this research, the story generator is aimed for children's education. Hence user interaction is a key component in order to maintain the child's interest in the generated stories. A new story generation approach which is user-driven is proposed. On top of that, the story generator will make use of existing public ontology as its knowledge base. This can help to reduce time and effort spent on developing the knowledge base of the story generator.

c. Develop the new solution

i. Here the focus is to study the selected public ontologies which are the WordNet and Animal ontology. Time is spent on developing parsers for these ontologies so that they can be used by the user-driven story generator. The proposed user-driven approach will be refined and developed in this stage as well.

d. Measure and analyze the new solution

i. After the story generator is developed, it is important to evaluate and compare it to other existing story generators. The user-driven story generator is compared to existing story generators based on several factors as seen in Section 5.

ii. A survey is designed to validate the results of this research. Using the survey, the developed story generator is compared to existing story generator by asking respondents to rate the story generators based on several key criteria. Among the key criteria are the level of user interaction, user-friendliness of the story generator, the quality of the generated story and the coherence of the sentences in the story

3.2 Research schedule

In this section, the proposed project timeline and the tasks required in order to complete this project are described. Firstly, the list of deliverables and tasks for this project are listed down. This can be seen in the following table:

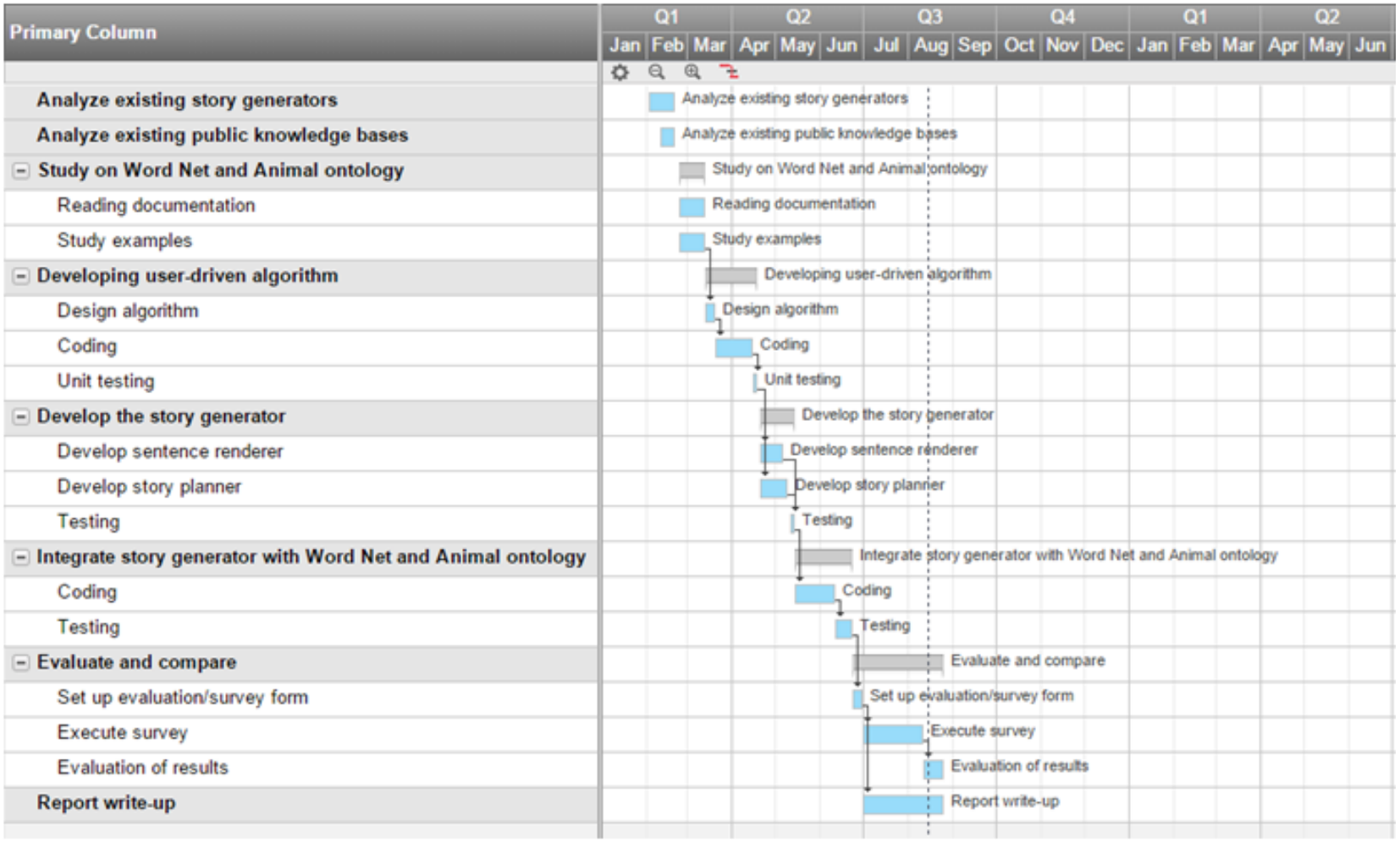
1.0 **Developing a user-driven story generator using public ontology**

- 1.1 Analyze existing story generators
- 1.2 Analyze available public ontologies
- 1.3 Study on Word Net and Animal ontology
  - 1.3.1 Reading documentation
  - 1.3.2 Analyze examples and APIs
- 1.4 Developing new user-driven algorithm
  - 1.4.1 Designing the algorithm
  - 1.4.2 Coding
  - 1.4.3 Testing
- 1.5 Developing the story generator
  - 1.5.1 Coding
  - 1.5.2 Testing
- 1.6 Integrating the story generator with WordNet and Animal ontology
  - 1.6.1 Coding
  - 1.6.2 Testing
- 1.7 Evaluating the usage of public ontology
  - 1.7.1 Evaluation
  - 1.7.2 Finalizing the story generator
- 1.8 Evaluate and compare the story generator
  - 1.8.1 Set up evaluation form
  - 1.8.2 Set up survey for comparison
  - 1.8.3 Survey
  - 1.8.4 Evaluation of results

Then the duration needed for each of the tasks is estimated and translated into a Gantt chart. The Gantt chart helps to give estimation on the timeline of this project.

Refer to the table below for the detail of each activity:

Primary Column	Start Date	End Date
Analyze existing story generators	02/02/15	02/19/15
Analyze existing public knowledge bases	02/10/15	02/19/15
Study on Word Net and Animal ontology	02/23/15	03/12/15
Reading documentation	02/23/15	03/12/15
Study examples	02/23/15	03/12/15
Developing user-driven algorithm	03/13/15	04/17/15
Design algorithm	03/13/15	03/19/15
Coding	03/20/15	04/14/15
Unit testing	04/15/15	04/17/15
Develop the story generator	04/20/15	05/13/15
Develop sentence renderer	04/20/15	05/05/15
Develop story planner	04/20/15	05/08/15
Testing	05/11/15	05/13/15
Integrate story generator with Word Net and Animal ontology	05/14/15	06/22/15
Coding	05/14/15	06/10/15
Testing	06/11/15	06/22/15
Evaluate and compare	06/23/15	08/24/15
Set up evaluation/survey form	06/23/15	06/29/15
Execute survey	06/30/15	08/10/15
Evaluation of results	08/11/15	08/24/15
Report write-up	06/30/15	08/24/15



Next, the Gantt chart is plotted using the information from the table. From the chart, it can be seen that the research takes around 6 months to be completed.

CHAPTER 4

FORMULATION OF USER-DRIVEN STORY GENERATOR BASED ON PUBLIC KNOWLEDGE BASE

As described in Section 2.1, a story generator mainly consists of three main components which are the story planner, sentence renderer and the knowledge base. This section discusses about the development stages of the story generator and the details of the main components. This includes the user-driven approach used in the story planner, the parsers for public knowledge base, the sentence renderer as well as the user-interface of the story generator.

4.1 Overview

In this research, a new user-driven approach for story generation is developed. The approach is user-driven, where response and answers given by the user will affect the story progression. This approach is implemented into the story planner component.

Another key part of this research is to make use of existing public knowledge bases. By making use of the existing public knowledge base, development time can be reduced as compared to developing a knowledge base from scratch. After analyzing various public knowledge bases, decision is made to make use of the Word Net and Animal ontology as the knowledge bases for this story generator.

In this research, a fixed plot for the story is used, with varying characters and actions depending on the user's input. Each time a user answers a given question; the story planner will retrieve the response and use it to determine the story progression and then interact with the sentence renderer to render the sentence(s) accordingly. The story

planner makes use of information from the knowledge base to determine the next sequence of questions as well as story progression. The diagram below describes the interaction between the sentence renderer, the story planner and the knowledge base.

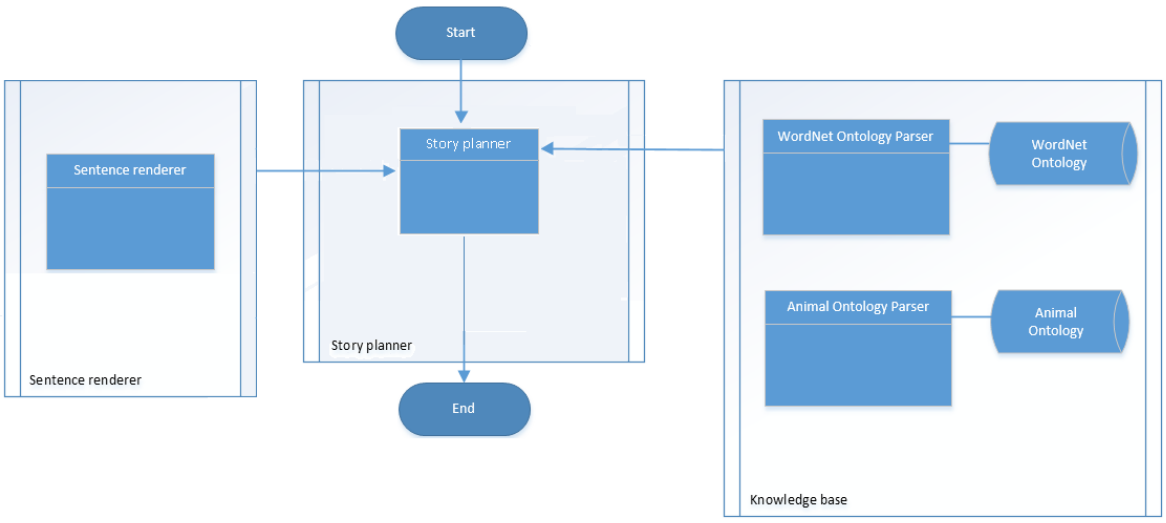


Figure 4.1: Interaction between key components of the user-driven story generator

4.2 Development Phases

In order to develop the user-driven story generator, each of the main components is developed separately and then integrated together to form the story generator. The flowchart below describes the development phases of this research. For the timeline of each development phase, please refer to Section 3.2.

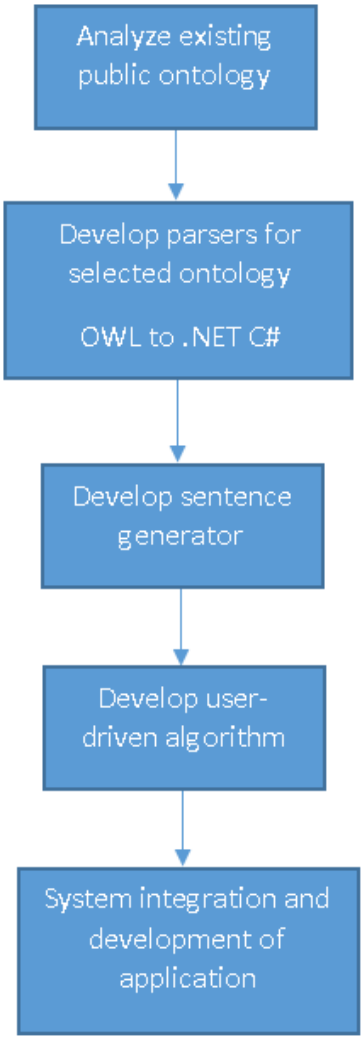


Figure 4.2: Development phases of user-driven story generator

4.3 Analyzing Word Net and Animal Ontology

4.3.1 Word Net

Word Net is a large lexical database of English which contains nouns, verbs, adjectives and adverbs (Princeton, 2015). The Word Net also includes relations between words such as hypernym, hyponym and meronym. In this research, the WordNet.OWL file is used as part of the knowledge base. The file used is version 3.0 and is fairly large with a size of more than 160 MBs. Loading the

WordNet.OWL file using open-source OWL-to-.NET parsers is not feasible as it takes up more than 10 minutes.

For this research, a parser is developed to retrieve required information from the Word Net and use it in the sentence renderer or story planner. Among the required information includes words of same meaning (which are "derived words with same meaning" or "synonyms"), adjectives, nouns and adverbs.

4.3.2 Animal ontology

The Animal ontology is in the OWL format. It describes animals and categorizes them based on their species, habitats, body parts, preferred food and types of movements. In the user-driven story generator, this ontology is used as part of the knowledge base and it provides the "character selections" for the story. Information of the selected animal such as the animal's habitat, food and movements are retrieved from the animal ontology. A parser is developed for this ontology.

4.4 Parsers

The Word Net ontology and the Animal ontology come in the form of OWL format. OWL is a semantic web language to represent complex knowledge and information of a given domain (Princeton, 2015). In order to read and retrieve data from these ontology files, a parser is needed in which it can extract the OWL file and return the data in the form of programmable language.

4.4.1 Word Net parser

There are open-source OWL-to-C# parsers available for public use such as OWLAPI. However, these parsers are very slow and take a very long time to load the Word Net file. This was tested with the OWLAPI and Protégé software, both taking up to more than 10 minutes to load the Word Net file. In order to parse the Word Net file, a parser has been developed from scratch for the Word Net file.

The table below describes some of the main functions of the developed Word Net parser:

- Word Net parser: OWL ontology → .NET (C#)

Function	**Description**
getNoun	Retrieves random nouns from Word Net
getVerb	Retrieves random verbs from Word Net
getAntonyms	Retrieves antonyms of a given word
getHypernyms	Retrieves hypernyms of a given word. Hypernym: Y is a hypernym of X if every X is a kind of Y. (canine is a hypernym of dog)
getEntailments	Retrieves entailments of a given word. Entailment definition: The verb Y is entailed by X if by doing X you must be doing Y (snore is an entailment of sleep)
getWordType	Returns the word type based on a given word ID
getRelatedAdverb	Get related adverbs of a given word
getDerivRelatedWords	Get similar nouns of a given word
getSimilarWords	Get similar words of a given word. Returns any word type, i.e. nouns, verbs or adverbs

4.4.2 Animal ontology parser

Similarly, a parser was developed for the Animal ontology. The table below describes some of the functions of the Animal ontology parser:

- Animal ontology parser: OWL ontology → .NET (C#)

Function	Description
getAnimalList	Get a list of animals of a given environment
getAnimalMovements	Get the possible types of movements for a given animal
getFoodList	Get the list of food of an animal
animalsInforest	Returns a list of animals which live in forest
animalsInSea	Returns a list of animals which live in sea

4.5 Sentence renderer

The sentence renderer is responsible for constructing sentences for the story generator. The story plot resides in the sentence renderer. The story generator developed in this research uses a fixed story plot with varying characters, objects and actions. The variables change accordingly depending on the user-input. The table below describes some of the functions of the sentence renderer:

Function	Description
generateIntro	Generates the introduction of the story based on the given user input
generateConflict1	Generates the first part of the story's conflict
generateConflict2	Generates the second part of the story's conflict
generateLesson	Generates the lesson of the story
getPlural	Gets the plural form of a given word
addVowel	Adds a vowel (a or an) to a given word
WasOrWere	Retrieves the tense suitable for a given word

4.6 User-driven approach

This section discusses in depth, on the approach used for the user-driven story generator.

4.6.1 The approach

The approach used in the story generator is fairly straight-forward. In the initialization stage, the story generator loads the Word Net ontology and the animal ontology files. Next, it prompts the user to select the environment and character for the story. Once these parameters are received, the story generator generates the introduction of the story.

The next stage is the iterative stage, where the story generator prompts questions for the user and then the given user input is used to determine the story progression. This iterative stage can be repeated a few times to promote more user-interaction during the story generation stage.

In the last stage, the story generator generates the conclusion for the story. This simple approach is described in the diagram below:

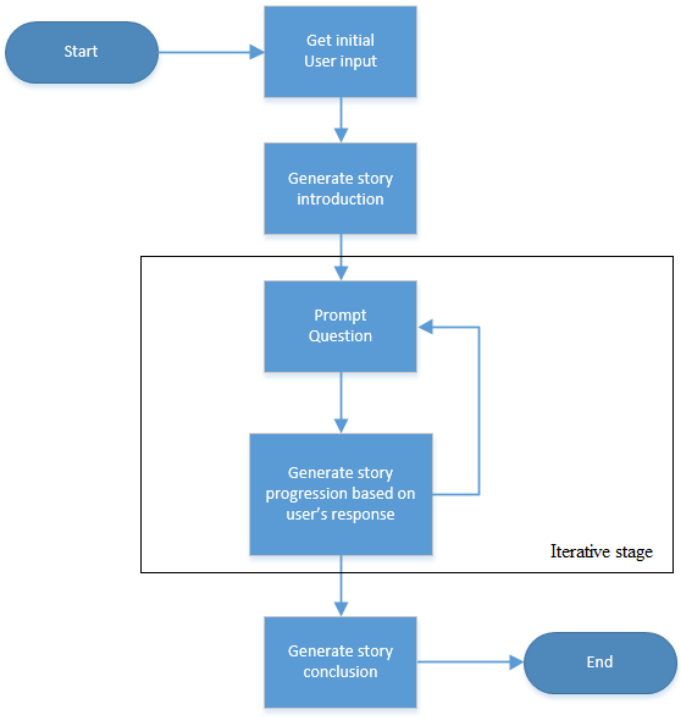


Figure 4.3: User-driven approach used in the story generator

4.6.2 Process flow

Based on the approach above, the story planner is developed. A simple story plot is used as the base of this story generator. The story plot is described below:

<Animal> lives in <habitat>. <Animal> is greedy. <Animal> loves to eat <food>.

<Animal> finds a lot of <food> in cave. Cave has small entrance. <Animal> <moves> towards cave and barely managed to enter it. <Animal> keeps eating the <food>. <Animal> can no longer exit the cave.

Lesson of the story is one should not be too greedy.

The words in bracket are replaced based on user's selection, where as words underlined are randomly replaced with synonyms. Replacing word with its random synonym is part of the story generator's randomization function to keep the generated story different each time. The diagram below gives an overview of the process flow of the story generator:

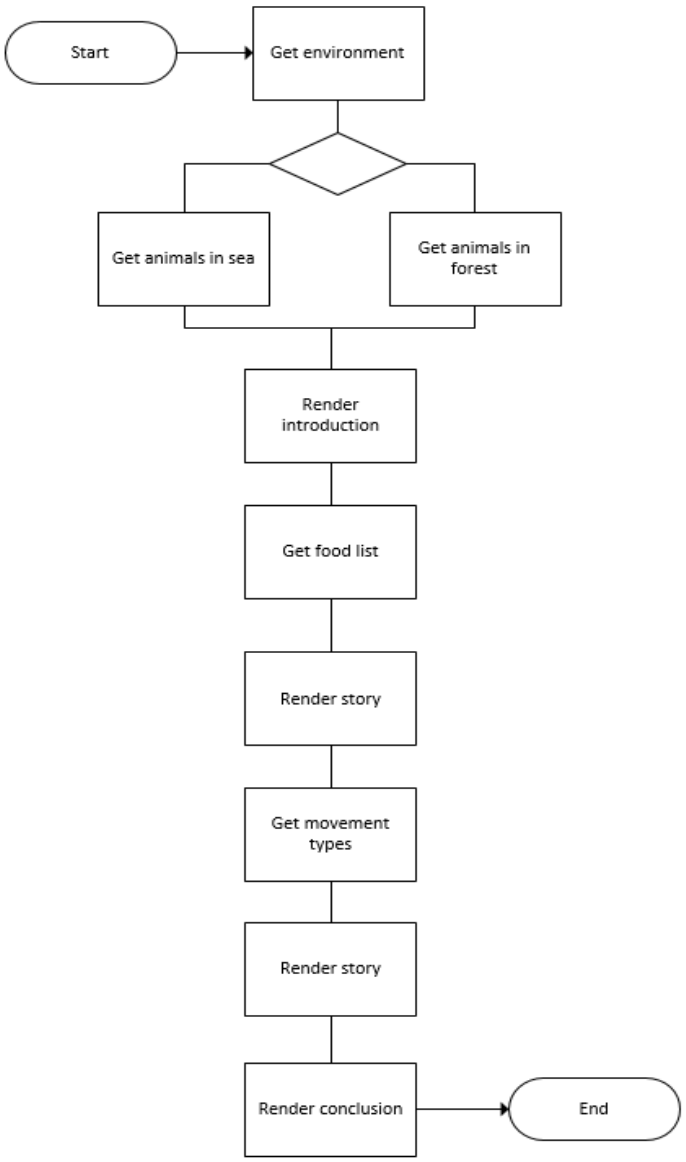


Figure 4.4: Process flow of the user-driven story generator

The application for the user-driven story generator will load the WordNet and Animal ontology files in the initialization stage. Next, it prompts user to select the "environment" for the story, which can either be forest or aquaculture. Based on user's selection, the story generator will then retrieve animals from the selected environment. After the user selects the animal, an introduction will be rendered. In the next step, the story generator finds the list of food of the selected animal and then, it prompts user to select a food. Next, the story will be appended

with generated sentences based on the selected input. After that, the story generator retrieves the list of movement types of the animal and prompts the user to select a type of movement. The story is then appended again with generated sentence based on the selected input. Lastly, the conclusion of the story is generated and displayed.

As can be seen from the process flow of the user-driven story generator application, the iterative stage of the approach (refer Figure 8) is only repeated twice, which are:

i) Get food list → Render story

ii) Get movement type → Render story

The application developed is a simple implementation of the user-driven story generation approach. For future improvement, the application can be programmed with more complex story plots and more iterations of the iterative stage. More future improvements for the user-driven story generator are discussed in Section 6.2.

4.7 Application

The story generator is developed using Microsoft Visual Studio 2013 and is written in the C# programming language. The "Windows Forms" is used to develop the user-interface and overall application of the user-driven story generator. For this user-driven story generator, there are basically two screens, which are the **Start screen** and the **Main screen**.

4.7.1 Start screen

The start screen has 2 input fields which allow users to select the path of the WordNet and Animal ontology files. After selecting the file paths, the user can click on the Start button to initialize the user-driven story generator. The diagram below shows the Start screen of the user-driven story generator:

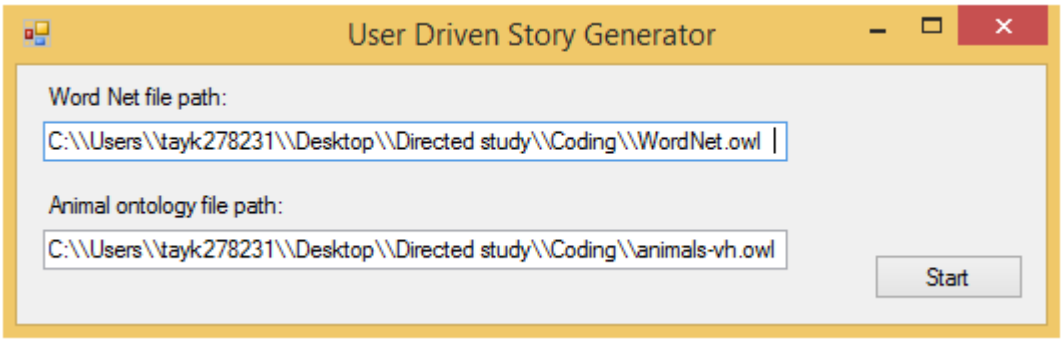


Figure 4.5: Start screen of the user-driven story generator

4.7.2 Main screen

The main screen has a huge panel on the right side which is called the story panel. The story panel displays the generated story. The main screen also has 3 buttons, which are the:

- **Next** button
    - User can click on this after selecting an answer for the question.
    - It will prompt the application to proceed with the next question and to generate sentences in the story panel
- **Reset** button
    - Resets the application and allow user to start the story again.
- **Exit** button
    - Closes and exits the application

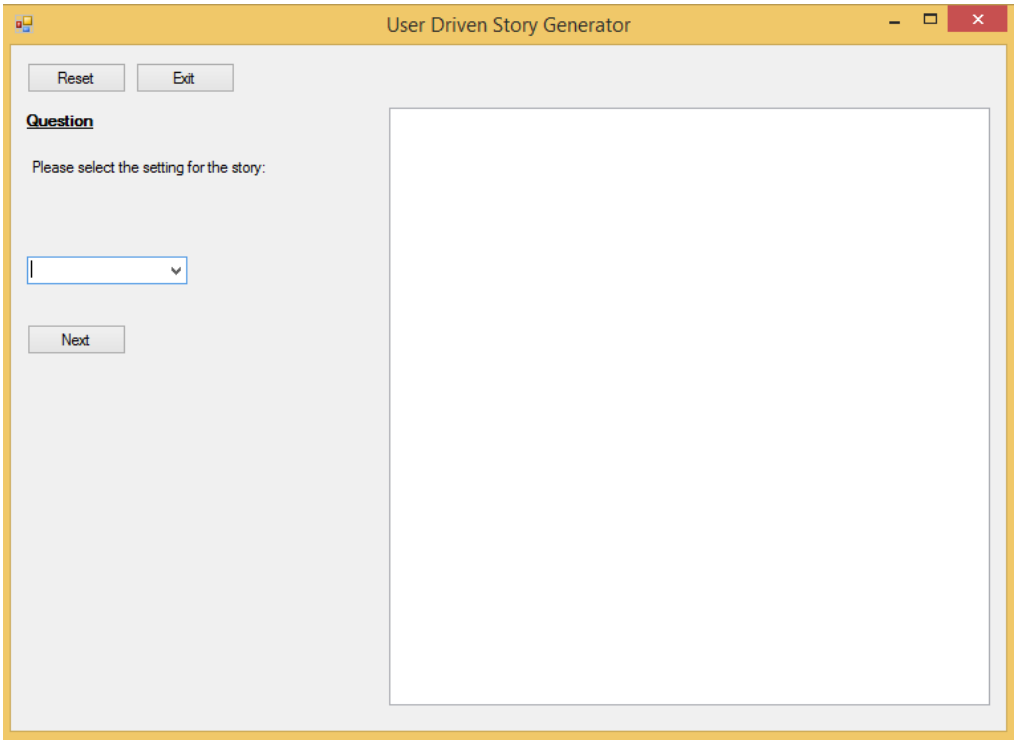


Figure 4.6: Main screen of the User-driven story generator

4.8 Significance of User-driven Story Generator Based on Public Knowledge Base

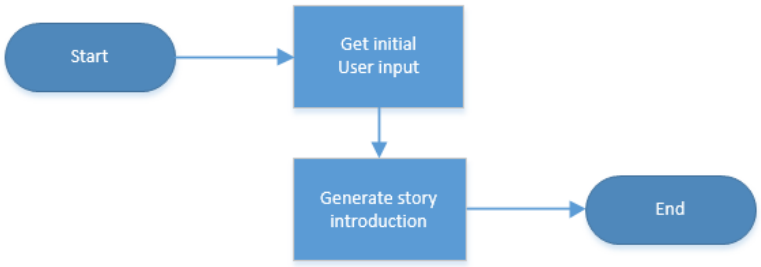


Figure 4.7: General process flow of existing story generators

The main objective of the user-driven approach used by the story generator is to promote more user interaction. Some of the existing story generators such as the data-driven story generator and the case-based story generator only receive user-inputs during the initial stage of story generation, refer to Figure 4.7 above. On the other hand, the user-driven story generator allows user to give responses even during the story generation process, refer to Figure 4.3 above. The user response will change and influence the progression of the story.

The story generator application implements the user-driven approach. In the diagram below, the story has already been generated half-way and it moves on to prompt another question to the user:

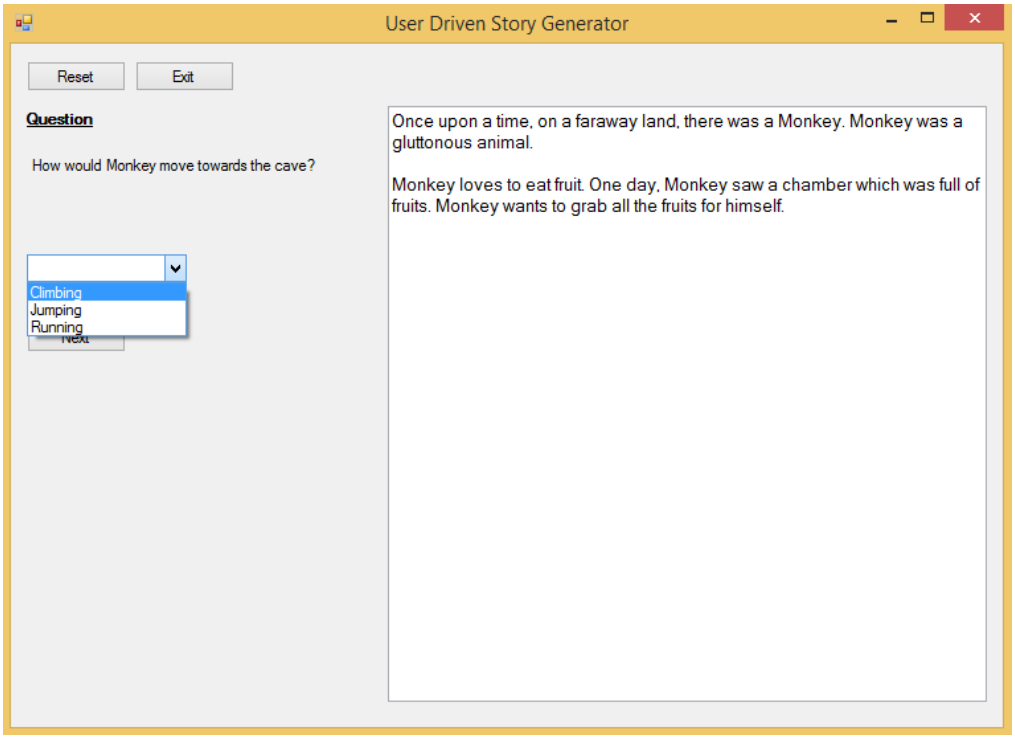


Figure 4.8: User-driven story generator – Prompt question to user

After selecting an answer, the story generator generates the story according to the user's response. This can be seen in the diagram below:

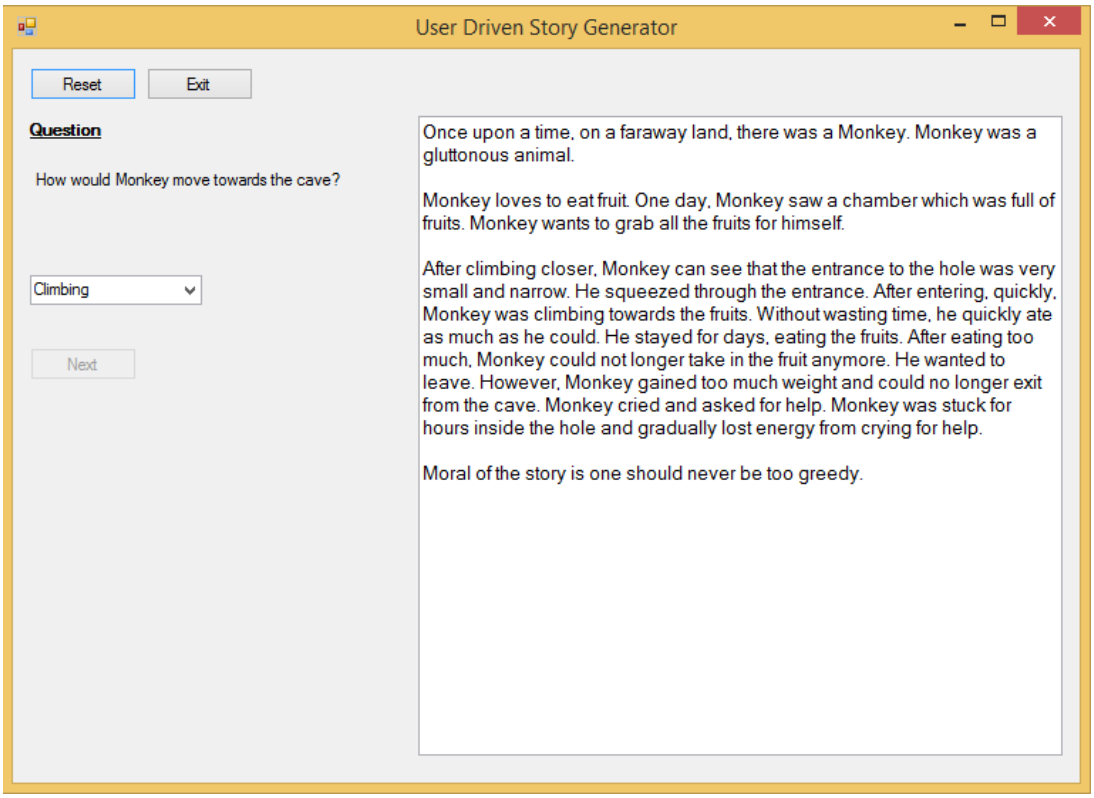


Figure 4.9: User-driven story generator – Story generator based on user's response

This approach can be applied and implemented into educational games for children. The approach allows children to think and provide responses which will then alter the generated story. This will directly help to trigger and stimulate the children's thinking process.

Another significance of the user-driven story generator is the implementation of existing public knowledge bases into the story generator. Instead of developing a knowledge base from scratch, a lot of development time can be saved by using an existing public knowledge base. In order to use an existing public knowledge base, the "bridge or parser" developed must be easily extensible and allows support for any extensions or enhancements of the public knowledge base.

In the user-driven story generator, the Word Net and Animal ontology are used and for example, currently in WordNet, the synonyms for the word “greedy” is only “gluttonous” and “voracious”. Hence the randomized words for greedy will either be “greedy”, “gluttonous” and “voracious”. The words returned are only limited to those 3 words. In the future, if the WordNet file is enhanced and more synonyms are added, the story generator can directly make use of these enhancements and added words. There is no need for additional development or coding effort. The same can be said for the Animal ontology. The story generator can directly reflect any added data to the Animal ontology, whether the added data are types of animals, types of movements, habitats or types of food.

4.9 Summary

This section described the formulation of the user-driven story generator, which includes each phase of the development phases. The development phases include the analysis and selection of existing public knowledge base, development of parsers for these knowledge bases, the formulation of the user-driven story generation approach, the development of the application and user-interfaces and lastly the integration of each component of the story generator.

CHAPTER 5

EVALUATION OF THE USER-DRIVEN STORY GENERATOR

This section discusses about the evaluation process of the user-driven story generator. Evaluation and comparison is done between the user-driven story generator and existing story generators. Next, in order to validate the user-driven story generator, a survey is designed and conducted. The survey tests and evaluates the user-driven story generator against the existing story generators such as the data-driven story generator and case-based story generator.

5.1 Comparison to existing story generators

The developed user-driven story generator is evaluated and compared to existing story generators based on the table below. The table compares the story generators based on the type of knowledge base used, the story generation approach, story plot pattern, user input parameters and also the type of user-interaction supported by the story generator.

	Story generators			
	User-driven story generator	**SUMOs**	**Data-driven generator**	**Case-based generator**
Knowledge base	Public ontology Word Net Animal ontology	Public ontology SUMO	Self-developed. Trained and grown using 437 fairy-tale stories	Self-developed ontology. Developed by breaking down existing stories into desired components.
Approach / algorithm	Based on user-driven approach	Based on initial user input to generate sentences	Score and rank method	Case-based reasoning. New stories are creating by replacing entities into an existing set of stories
Story plot pattern	Fixed story plot	Problem, rising action, climax and solution	Flexible and dynamic	Depending on the selected story
Initial user input	Characters, roles, places	Entities and domain	Topic and length of story	Characters, roles, places
User-interaction	Allows user to select input parameters Allows user to influence story progression during story generation process	Allows user to select input parameters by selecting pictures in the software (during initialization stage).	Only takes in input parameter during initialization stage, before story generation process begins.	Only takes in input parameter during initialization stage, before story generation process begins.

5.2 Designing the survey

The main purpose of this survey is to evaluate and compare the user-driven story generator to existing story generators which are the data-driven story generator and case-based story generator. Several criteria such as level of user interaction, ease-of-use and quality of the generated story in terms of creativity and coherence will be used for comparison. The result of this survey will determine if the null hypothesis (H_0) is accepted or rejected. The null hypothesis of this research is as described below:

H_0 = The user-driven story generator provides better user-interaction as compared to the data-driven story generator and case-based story generator.

5.2.1 Subject

The subjects (survey respondents) of this evaluation are required to use the selected story generators (refer Section 5.2.2) to generate stories and answer a series of questions (refer Section 5.2.3). Among some of the selected respondents are parents or children who are already schooling. The table below shows the list of respondents selected for this survey.

5.2.2 Subject system

The selected systems for this survey are the user-driven story generator, the data-driven story generator and the case-based story generator. The domain of these story generators is based on children fairy tales.

5.2.3 Questionnaire

The questionnaire for the survey consists of 2 sections. In Section A, the respondents' background and personal information are gathered.

Proposed questions for Section A:

No.	Question
A1	What is your age?
A2	What is your gender?
A3	What is your occupation?
A4	Select your preferred genre of stories
A5	Is reading one of your favorite pastimes?

In Section B, the respondents are required to rate and compare the story generators based on the given criteria. A scale of 1-to-5 will be used to score the story generators for each respective question, whereby a score of 5 is the highest and 1 is the lowest.

Proposed questions for Section B:

No.	Question
B1	Rate the story generators based on user-interaction
B2	Rate the story generator based on ease-of-use
B3	Rate the story generator based on story variation and diversities
B4	Rate the generated story based on plot development and creativeness
B5	Rate the generated story based on grammatical correctness
B6	Rate the generated story based on paragraph coherence

In the last section, the respondents are also requested to give some comments or possible improvements for the user-driven story generator. These comments and feedbacks from the respondents can be as reference for future enhancements of the user-driven story generator.

5.2.4 Procedure

In the survey, the respondents are asked to explore all 3 story generators which are mentioned in Section 5.2.2. They will generate a few stories using the story generators and move on to answer the survey questions. The steps below describe the procedure of this survey:

a) Explain and describe the purpose of the survey.
b) Briefly describe each of the story generators to the subjects.
c) Demonstrate once on how each story generator is used.
d) Allow the subjects to start exploring and using the story generators.
e) Request subjects to fill in the survey questionnaire.

5.3 Survey results

The survey results were collected from 17 respondents. An overall result of the survey is shown in the graph below.

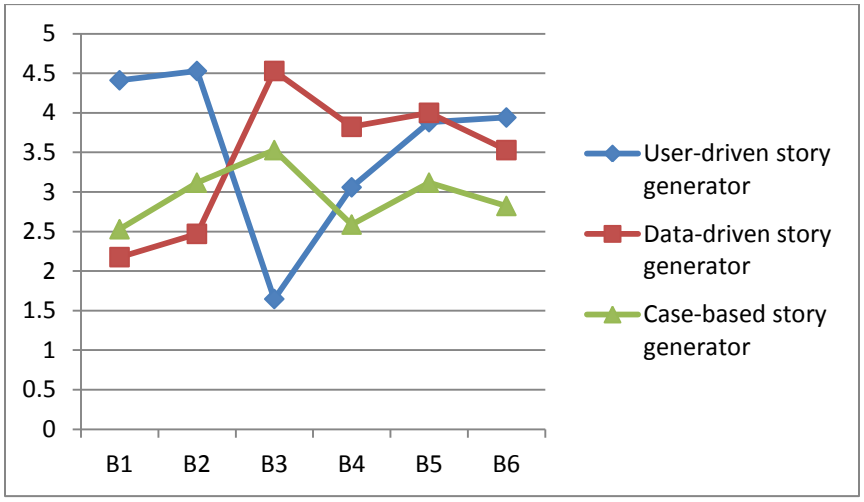


Figure 5.1: Average score for each question in Section B

From the results, it can be seen that the user-driven story generator scored the highest for question B1, which measures the level of user interaction of the story generator. The user-driven story generator scored an average 4.4 for question B1 as compared to the data-driven story generator (2.2) and case-based story generator (2.5).

The user-driven story generator also faired significantly better than the other 2 story generators for question B2, which measures the ease-of-use of the story generator. For question B2, the user-driven story generator has a score of 4.5, the data-driven story generator scored 2.5 and the case-based story generator scored 3.1.

On the other hand, the results for question B3, which measures the story variations and diversities, show that the user-driven story generator has a lower score as compared to the data-driven and case-based story generators. This can be explained because the user-driven story generator is only using a fixed story plot and there is only one story added to it prior to the survey.

For question B4, which measures the creativity of stories, the data-driven story generator has the highest score (of 3.8) as compared to the user-driven story generator (3.1) and the case-based story generator (2.6).

The user-driven story generator and data-driven story generator has similar scores for question B5 and B6, where as the case-based story generator scored lower for these 2 questions. A more detailed analysis on the results for each of the question in Section B is discussed in the following section.

5.3.1 Results analysis

In this section, the collected results are studied and analyzed based on each question in Section B.

Question B1: Rate the story generators based on user-interaction

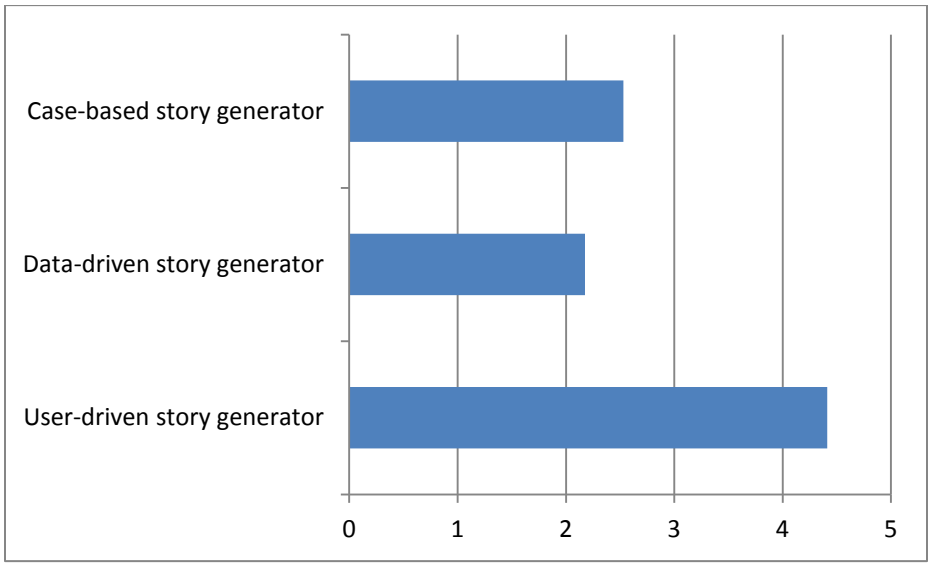


Figure 5.2: Average score for question B1

The user-driven story generator has the highest score for this question, scoring an average of 4.4 out of 5, as compared to the data-driven story generator (2.2) and the case-based story generator (2.5).

This can mainly be explained by the nature of the algorithms or approaches used by each of the story generator. For the data-driven story generator and data-driven story generator, the users are only allowed to select some predefined user parameters and the stories will be generated based on the selected input parameters. On the other hand, the user-driven story generator progressively prompts the user questions and given user input will affect the story progression. This allows more user-interaction where user is allowed to influence or change the generated story along the story generation process.

Question B2: Rate the story generator based on ease-of-use

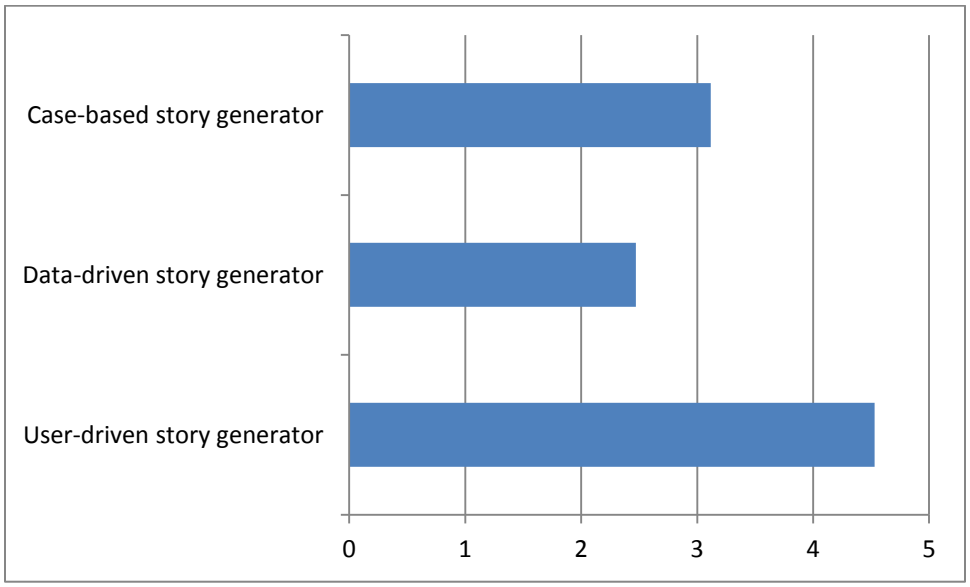


Figure 5.3: Average score for question B2

Question B2 asks the respondents to rate the ease-of-use of each story generator. The user-driven story generator has the highest average score (4.5) as compared to the data-driven story generator (2.5) and case-based story generator (3.1). The user-interface created for the user-driven story generator is very simple and easy to understand. The user can easily select answers for each prompted

question at the Main screen and the generated story is displayed in the same screen on a panel (refer to Section 4.7.2). On the other hand, the data-driven story generator requires user to fill in many parameters before the story can be generated (refer to Figure 17 below). The case-based story generator has many parameters for the user to select as well and it does not come with an explanation on what the parameters are for.

Figure 5.4: Input parameters for the data-driven story generator

The user interfaces of the data-driven and case-based story generators are less user-friendly and users find it harder to use as compared to the user-driven story generator.

Question B3: Rate the story generator based on story variation and diversities

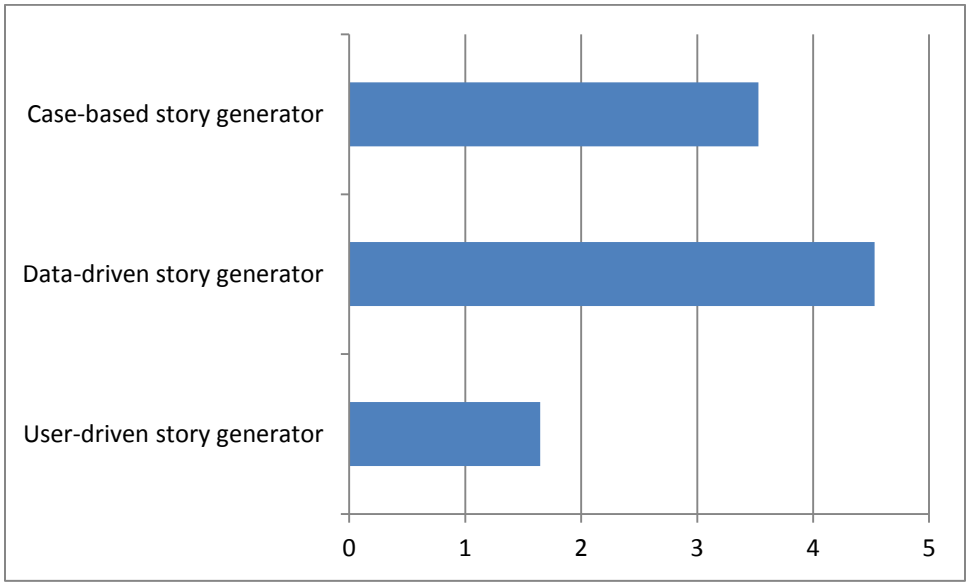


Figure 5.5: Average score for question B3

Question B3 rates the variations and diversities of the generated stories by the story generators. For this question, the user-driven story generator has the lowest score (1.6), as compared to the data-driven story generator (4.5) and the case-based story generator (3.5). The data-driven story generator and case-based story generator has a larger knowledge base and are able to generate many different stories based on the given user inputs. However, the user-driven story generator developed for this research is only using a fixed story plot and there is only one story added to this generator. Hence the user will easily find the story generated by the user-driven story generator to be repetitive and boring.

Question B4: Rate the generated story based on plot development and creativeness

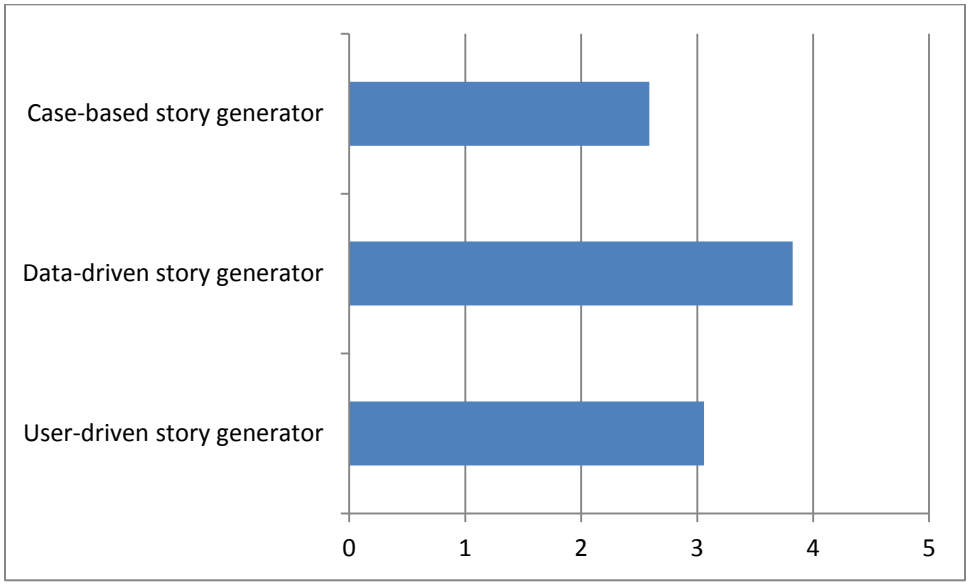


Figure 5.6: Average score for question B4

For Question B4, the data-driven story generator scored higher (3.8) than the user-driven story generator (3.0) and the case-based story generator (2.6). This is because the data-driven story generator requires a lot of user-input and these user inputs are used to help generate many random sentences which can be randomly joined to form a story. This allows the generated story to have more depth and creativity as compared to the user-driven story generator and case-based story generator, where both story generators are using fixed story plots.

Question B5: Rate the generated story based on grammatical correctness

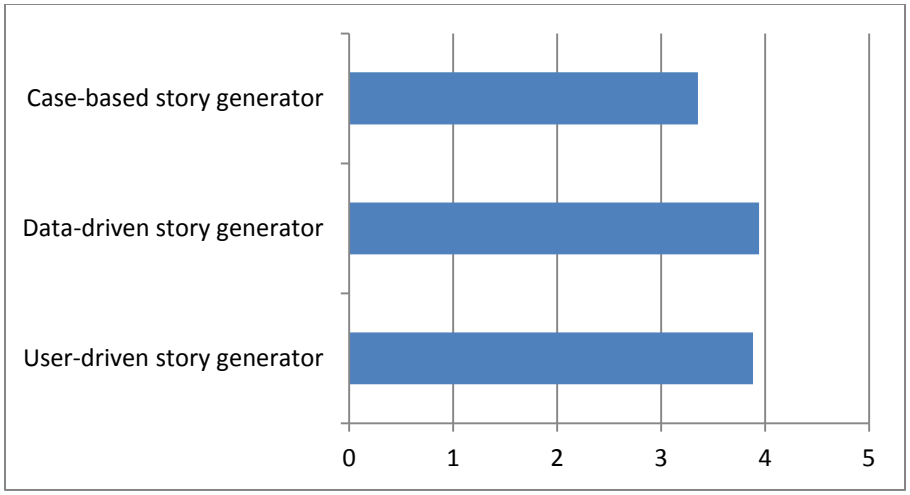


Figure 5.7: Average score for question B5

Question B5 measures the grammatical correctness of the sentences generated by each of the story generator. For this question, all three story generators have similar scores. There are generated stories all have low grammatical errors hence the respondents have rated all three story generators to with similar scores of around 3.3 to 3.8.

Question B6: Rate the generated story based on paragraph coherence

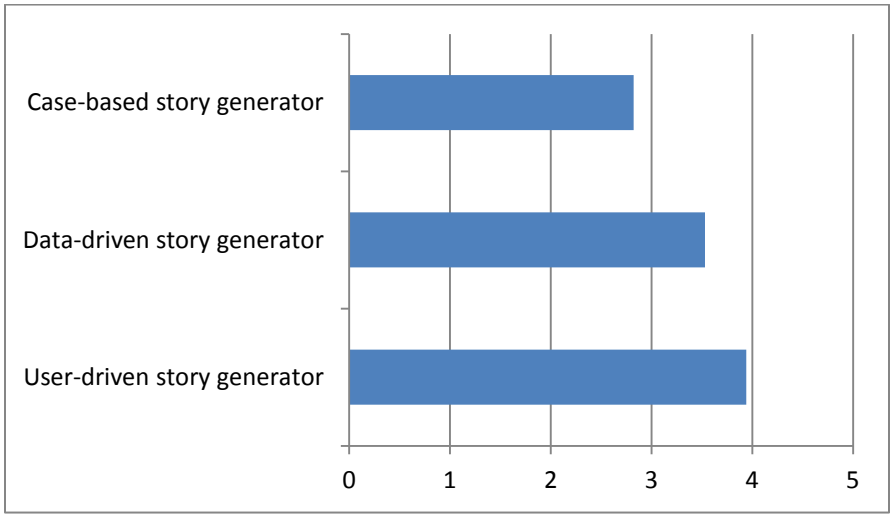


Figure 5.8: Average score for question B6

Question B6 asked the respondents to rate the paragraph coherence of the generated stories. Coherence in a paragraph is the technique of making words, phrases, and sentences move smoothly and logically from one to the other. For this question, the case-based story generator scored the lowest (with 2.8), as compared to the data-driven story generator (3.5) and user-driven story generator (3.9). This is because the user-driven story generator uses a fixed story plot and only replaces characters, object and some verbs in the story. Hence the story's original plot and story progression is not affected, thus having higher paragraph coherence. On the other hand, the data-driven story generator also has very low randomization factor as it receives a lot of user inputs for its story generation.

As for the case-based story generator, it uses many story plots and mixes them together. In many cases, the generated stories are not coherent and do not have a proper story progression or story flow. The figure below shows an example of story generated using the case-based story generator, where the story is actually a combination of 2 different story frames and there is no continuity between the story frames. This caused a low coherence between the paragraphs of the stories.

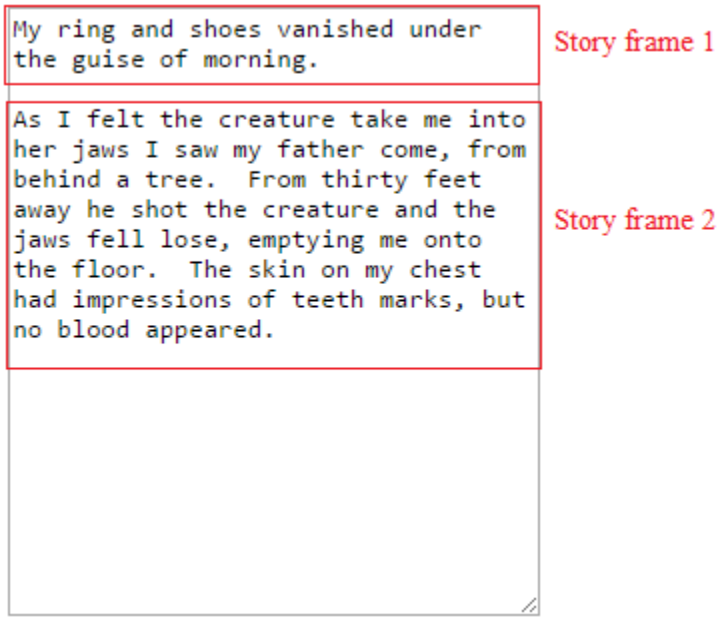


Figure 5.9: Story generated using case-based story generator

5.4 Analyzing validity threat

The survey results may not be accurate due to the small sample size. The number of survey responses obtained from 17 respondents, hence not providing a better statistical analysis of the results. Besides that, some of the other threats to the validity of the survey include:

a) Bias in survey response from subjects of similar background

Most of the selected respondents are working professionals in the field of Information Technology. As they come from similar education and working background, the responses given by them may be biased and are not diverse enough. This can be prevented by gathering a larger sample size and get more respondents from different background.

b) Survey is conducted personally

When conducting the survey personally, the subjects may be influenced by the survey conductor to give better scores for the user-driven story generator. The survey can be conducted by a person who is not involved in this research in order prevent respondents being influenced or swayed into giving responses which may benefit the user-driven story generator.

5.5 Summary

This chapter discusses the evaluation done for the user-driven story generator. The proposed null hypothesis for this research is to determine whether the user-driven story generator allows more user interaction when compared to existing story generators such as the case-based story generator and data-driven story generator. A survey was designed to test this and there were a total of 17 respondents who have participated in the survey. Based on the survey results, the user-driven story generator proves to have a better level of user-interaction and is more user-friendly when compared to the existing story generators such as the data-driven story generator and case-based story generator. However, the survey results also showed that the user-driven story generator lacks in the area of story creativity and the generated stories are too repetitive.

CHAPTER 6

CONCLUSION

This chapter presents the summary of the research, the research contributions, as well as possible future enhancements of the user-driven story generator.

6.1 Research summary

The objectives of this research are as described in Section 1.3. These objectives are achieved in this research. The objectives are as follow:

a) To analyze existing story generation algorithms and approaches

In this research, several existing story generation algorithms such as the data-driven story generation algorithm, case-based algorithm and role-based approach are studied and analyzed. A thorough literature review was done for these story generation algorithms. The strengths and weaknesses of these story generation algorithms can be seen in Section 2.3.

b) To formulate and design a new user-driven story generation approach

One of the main weaknesses of existing story generators is the lack of user interaction. Hence in this research the aim is to develop a new user-driven story generation approach which allows for more user interaction. The developed user-driven story generation approach allows users to influence and change the

generated story even during the story generation process. The user is progressively prompted with questions during the generation of story. The given answer is used by the story generator to determine the story generation and next sequence of questions. More detailed explanation of this user-driven story generation approach is found in Section 4.6. In this research, two existing public knowledge bases in the form of ontology are used. The selected ontologies are the Word Net ontology and the Animal ontology. The Word Net ontology has a huge base of English words such as nouns, adjectives, synonyms, hypernyms and relations between words. This knowledge base is used to generate sentences and randomize words in the user-driven story generator. The Animal ontology consists of data on many types of animals. It contains information on the animals' habitat, movement types, preferred food and also their body parts. In the user-driven story generator, the Animal ontology is used to provide the environment, animal characters and also the animals' list of food and movement types. In order to integrate these knowledge bases into the user-driven story generator, a parser is built for each of the knowledge bases.

c) To develop and implement the user-driven story generation approach into the story generator

A story generator application is developed in this research. This story generator implements the user-driven story generation approach and also incorporates existing public ontologies which are the WordNet and Animal ontology. The developed application is described in Section 4.7.

d) To evaluate the user-driven story generator

A survey is done to evaluate and compare the user-driven story generator with the existing story generator such as the data-driven story generator and the case-based story generator. The survey results show that the user-driven story generator has a higher level of user-interaction as well as better ease-of-use as

compared to the data-driven story generator and case-based story generator. Refer to Section 5.0 for more information on the survey design, survey procedure and survey results.

6.2 Future enhancements

The user-driven story generator can be further improved. Some of the possible improvements and enhancements are as listed below:

i) **Provide story twists based on user's input**. For example, refer to the questions below and the corresponding generated sentence.

Scenario 1

- **Question**: How would the monkey move towards the cave?
- **User input**: Climbing
- **Sentence output**: Monkey successfully climbed towards the cave and entered it.

Scenario 2

- **Question**: How would the monkey move towards the cave?
- **User input**: Walking
- **Sentence output**: Monkey walked towards the cave and stepped on a trap. He was caught.

With story plot twists in the generated story, the users will find the stories to be more creative and interactive as the story line may change depending on the selected response.

ii) Adding more stories to the story generator

From the survey respondents, there are several feedbacks which requested the user-driven story generator to provide more stories. Currently the user-driven story generator only has one story, and this will cause the users to feel bored with the story generator because the generated stories are repetitive.

iii) Improve user interface by adding pictures

The story generator's target audience will be children. Having more pictures to describe the characters or the story's background will help to attract children's attention

REFERENCES

Meehan, James. 1977. **An interactive program that writes stories**. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence.* Cambridge, Massachusetts, pages 91-98

McIntyre, Neil and Mirella Lapata. 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.* Singapore, pages 217-225

McIntyre, Neil and Mirella Lapata. 2010. **Plot induction and evolutionary search for story generation**. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Uppsala, Sweden, pages 1562-1572.

Chen, Sherol, Adam M. Smith, Arnav Jhala, Noah Wardrip-Fruin and Mateas Michael. 2004. **RoleModel: Towards a Formal Model of Dramatic Roles for Story Generation**. University of California Santa Cruz, USA.

Cua, Jeffrey, Ong, Ethel, Pease, Adam, and Manurung, Ruli. 2010. **Representing Story Plans in SUMO**. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity.* Los Angeles, California, pages 40-48.

A.Jaya and G.V. Uma. 2010. **An intelligent system for semi-automatic story generation for kids using ontology**. Anna University, Chennai, India.

Pease, A. 2006. **Formal Representation of Concepts: The Suggested Upper Merged Ontology and Its Use in Linguistics. Ontolinguistics. How Ontological Status Shapes the Linguistic Coding of Concepts**. Schalley, A.C. and Zaefferer, D. (ed.), Berlin New York.

Pease, A . 2013. **Suggested Upper Merged Ontology (SUMO)**. Retrieved March 1st, 2014 from http://www.ontologyportal.org/index.html

Machado, J. 2003. Storytelling. In **Early Childhood Experiences in Language Arts: Emerging Literacy**, pages 304-319. Clifton Park, New York, Thomson/Delmar Learning.

Stanford University. *Animal ontology*. Retrieved March 12th 2015, from protege.stanford.edu/junitOntologies/testset/animals-vh.owl

Gervas, P., Diaz-Agudo, B., Peinado, F. & Hervas, R. 2004. **Story Plot Generation Based on CBR**. Faculty of Informatics, Complutense University of Madrid, Madrid, Spain.

Basili, V.R., The Experimental Paradigm in Software Engineering. International Workshop, 1992.

Marcos., M.L.a.E., Research in Software Engineering: Paradigms and Methods, in In Proceedings of the 17th International Conference on Advanced Information System (CAiSE'05). June 2005: Porto, Portugal.

Hong, A., Solis, C., Siy, J.T., and Tabirao, E. 2008. *Picture Books: Automated Story Generator*. Undergraduate Thesis, De La Salle University, Manila, Philippines.

Princeton University, March 2015. *Word Net*. Retrieved August 20th 2015 from https://wordnet.princeton.edu/

Pengurusan Sumber Pendidikan. *Program NILAM*. Retrieved August 12th 2015, from http://bibliografi.moe.edu.my/SumberPendidikan/index.php/faqs/214-program-nilam.html

APPENDIX A

University Tunku Abdul Rahman

Master in Information Systems

A Survey on Existing Story Generators

versus

User-driven Story Generator Using Public Knowledge Base

Story generators help in generating stories automatically which can be used in either movies, games or even in education. There are various kinds of existing story generators, each using its own story-generating algorithm. In our research, we have created a user-driven story generator which is based on public knowledge base.

In this survey, we would like to obtain your opinions regarding the user-driven story generator and also to compare it to existing story generators.

Objective of the survey

To compare the existing story generators with the user-driven story generator.

Survey structure

Section A Personal background

Section B Evaluation of the story generators

It should take approximately 20-30 minutes to complete the questionnaires. We would like to plead for sincere participation and your cooperation in answering these questionnaires is very much appreciated. Thank you very much.

Section A: Personal Background

A1: What is your age?

	Under 13
	13-19
	20-30
	30 and above

A2: What is your gender?

	Male
	Female

A3: What is your occupation?

_____

A4: Of the categories below, select your preferred genre of stories.

	Action
	Adventure
	Comedy
	Fantasy
	Horror
	Romance
	Others

A5: Is reading one of your favorite pastimes?

	Yes
	No

Section B: Evaluation of the story generators

In this section, you need to interact with the following story generators in order to generate a story:

User-driven story generator

Data-driven story generator (http://www.plot-generator.org.uk/)

Case-based story generator (http://tz69.3x.ro/Generator/gen.html)

B1: Rate the story generators based on user-interaction

5 – Very interactive

4 – Interactive

3 – Normal

2 – Less interactive

1 – Not interactive

User-driven generator	
Data-driven generator	
Case-based generator	

B2: Rate the story generator based on ease-of-use:

5 – Very easy to use

4 – Easy to use

3 – Normal

2 – Complicated

1 – Very complicated

User-driven generator	
Data-driven generator	
Case-based generator	

B3: Rate the story generator based on story variation and diversities (please generate a few stories using each of the story generators):

5 – Very diverse

4 – Diverse

3 – Normal

2 – Repetitive

1 – Very repetitive

User-driven generator	
Data-driven generator	
Case-based generator	

B4: Rate the generated story based on plot development and creativeness:

5 – Very good

4 – Good

3 – Normal

2 – Poor

1 – Very poor

User-driven generator	
Data-driven generator	
Case-based generator	

B5: Rate the generated story based on grammatical correctness:

5 – Very good

4 – Good

3 – Normal

2 – Poor

1 – Very poor

User-driven generator	
Data-driven generator	
Case-based generator	

B6: Rate the generated story based on paragraph coherence.

(Note: ***Coherence*** in a ***paragraph*** is the technique of making words, phrases, and ***sentences*** move smoothly and logically from one to the other.)

5 – Very good

4 – Good

3 – Normal

2 – Poor

1 – Very poor

User-driven generator	
Data-driven generator	
Case-based generator	

Remarks / comments (if any):

_____

_____

_____

_____