



Instituto Politécnico Nacional

Centro de Investigación en Computación

Automatic text generation by learning from literary structures

T E S I S

Que para obtener el grado de:
Maestría en Ciencias de la Computación.

PRESENTA:

Ing. José Angel Daza Arévalo

Directores de Tesis:

Dr. Francisco Hiram Calvo Castro
Dr. Jesús Guillermo Figueroa Nazuno

MÉXICO, D.F.

DICIEMBRE 2015





SIP-14 bis

INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 13:45 horas del día 12 del mes de noviembre de 2015 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

Centro de Investigación en Computación

para examinar la tesis titulada:

"Automatic text generation by learning from literary structures"

Presentada por el alumno:

DAZA

Apellido paterno

ARÉVALO

Apellido materno

JOSÉ ANGEL

Nombre(s)

Con registro:

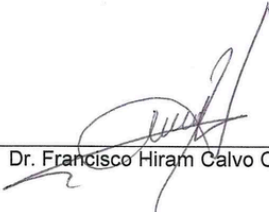
B	1	3	0	0	7	5
---	---	---	---	---	---	---


aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**


Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISIÓN REVISORA

Directores de Tesis

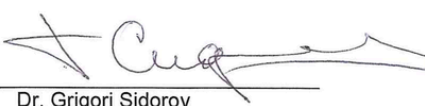

Dr. Francisco Hiram Calvo Castro


Dr. Jesús Guillermo Figueroa Nazuno


Dr. Sergio Suárez Guerra


Dr. Alexander Gelbukh


Dra. Olga Kolesnikova


Dr. Grigori Sidorov

PRESIDENTE DEL COLEGIO DE PROFESORES


Dr. Luis Alfonso Villa Vargas

ESTADOS UNIDOS MEXICANOS
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA CESIÓN DE DERECHOS

En la Ciudad de México D.F. el día 30 del mes Noviembre del año 2015, el (la) que suscribe José Angel Daza Arévalo alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro B130075, adscrito al Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Francisco Hiram Calvo Castro y Dr. Jesús Guillermo Figueroa Nazuno y cede los derechos del trabajo intitulado Automatic text generation by learning from literary structures, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección Av. Juan de Dios Bátiz, Esq. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, Delegación Gustavo A. Madero, C.P. 07738, México D.F. o al correo electrónico josdaza.a@gmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

José Angel Daza Arévalo
Nombre y firma

RESUMEN

El lenguaje escrito es uno de los inventos más importantes de la humanidad. Gracias a él, ha sido posible comunicar ideas complejas a través del tiempo. De igual forma, la invención de las computadoras cambió por completo la manera en que los seres humanos nos comunicamos y expresamos ideas. Sin embargo, la tarea de generar automáticamente lenguaje humano ha resultado mucho más difícil de lo que originalmente se pensaba. En general, conceptos fuertemente ligados a la inteligencia humana tales como el arte, la creatividad y la creación literaria apenas comienzan a ser tomadas en cuenta desde una perspectiva automática debido a los problemas que esto conlleva, tales como el adecuado manejo del significado, la intencionalidad, la capacidad de planeación y el poseer sentido común, sólo por nombrar algunos.

En este trabajo proponemos una nueva metodología para la generación automática de textos de ficción. Nuestra intención es encontrar una manera adecuada de formalizar el proceso de escritura creativa; por lo tanto, creemos que es posible definir un método simplificado capaz de automatizar dicho proceso. Basándonos en la definición de tal método, seremos capaces de generar textos de ficción que contengan oraciones originales y coherentes, mediante la adaptación de contenido presente en textos previamente dados.

El trabajo actual persigue la generación de textos originales basándose en la combinatoria dirigida, explotando únicamente patrones sintácticos y semánticos que se encuentren en lo que ha sido escrito ya. Intentamos emular a los autores humanos en el sentido de que cada uno de nosotros, como autores, nunca producimos un texto nuevo a partir de la nada, sino que aplicamos conocimiento previamente adquirido a partir de la experiencia o la lectura y asimilación de textos ajenos.

La mayoría de los trabajos previos en esta área intentan formalizar la generación de texto desde una perspectiva que va de lo general a lo particular, donde la meta principal es planear la idea global a comunicar, y la producción lingüística es relegada al final, cumpliendo únicamente con la tarea de llenar estructuras previamente armadas. Por otro lado, una de las principales contribuciones de este trabajo es la propuesta de una perspectiva que va de lo particular a lo general, donde una sola palabra es el punto de partida, y el significado del texto final será un producto del significado potencial que pueda emerger desde dicha palabra inicial al momento de realizar la producción lingüística; y es solamente al final en donde una estructura general se aplica al texto con la intención de presentarlo como una historia.

Otro aspecto importante que es explorado en este trabajo es la creatividad. Al dejar que el significado del texto emerja automáticamente, sin imponer una intención determinada desde el inicio, las historias generadas pueden ser llamadas creativas, pues el texto contiene una serie de significados que no necesariamente estaban presentes en el corpus de texto usado para el

aprendizaje. Para probar esto, demostramos que los evaluadores humanos consideran nuestros textos al menos igualmente creativos que los textos creados explícitamente por un autor humano.

La intención de este trabajo en general es demostrar que el significado puede emerger a través de propiedades implícitas en el uso del lenguaje, y que no necesariamente es la intencionalidad la única capaz de dotar de significado a un texto. Lo que proponemos es derribar el mito de la creatividad y el arte como actividades únicamente humanas, de la misma forma en que Turing liberó la definición de inteligencia del pensamiento meramente humano. Al hacer esto, esperamos motivar un interés más profundo en el estudio de la creatividad y las máquinas.

ABSTRACT

Written language is one of the most important inventions of humanity; because of it, complex ideas were able to be communicated through time. In the same manner, the invention of computers completely changed the way human beings communicate and express ideas. However, the task of generating human language automatically has proved to be much more difficult than it was expected at the beginning. In general, strongly tied concepts to human intelligence such as art, creativity and storytelling are just beginning to be seriously explored with an automatic approach because of the problems it triggers, such as dealing with meaning, intentionality, planning and common sense knowledge just to mention a few.

In this work we propose a new methodology for automatic fiction text generation. We intend to achieve this by attempting a formalization of the creative writing process; thus, we believe that we can define a simplified method to automatize the creative writing process. With such a method, we will be able to generate fiction texts that contain novel and coherent sentences by adapting them from previous fiction texts.

The present work tries to generate original and novel texts based on a directed combinatorial perspective by exploiting syntactic and semantic patterns found in what has already been written. We attempt to emulate human authors in the sense that we as authors do not produce a new text from a fresh start, but by applying previous knowledge acquired from what we have already read and experienced.

Many of the previous works from this area try to formalize a general top-down text generation technique, where overall text planning is the main goal, and linguistic production is left to the end only to fill the already given structured statements. On the other hand, one of the main contributions of this work is the bottom-up approach, where a single word is the starting point, and the meaning of the generated text emerges automatically together with linguistic production; and only in the end, a general structure will be given to the text in order to be able to present it as a story.

Another important aspect that we explore on this work is creativity. As we let text meaning emerge instead of determining it from the beginning, the generated stories can be called creative, since the text hold a series of meanings that are not necessarily present in the corpus. To prove this, we demonstrate how human evaluators consider our texts as creative as those which were written explicitly by a human author.

On the whole, the intention of this work is to demonstrate that meaning can emerge through language properties and usage, and not necessarily from intentionality. We intend to demystify the idea that creativity and art are necessarily tied to a human artist, just as Turing detached the definition of intelligence from human thought. By doing this, we expect to encourage a more profound interest on the study of machines and creativity.

AGRADECIMIENTOS

*A mi familia, en especial a mis padres y mis hermanas,
por su presencia, apoyo y ejemplo:
inagotables, incondicionales, inconmensurables.*

*A mis directores de tesis, por sus valiosos consejos
y por creer en este proyecto desde un inicio.*

*A mis amigos, los nuevos y los eternos,
que hacen de éste el mejor de mis mundos posibles.*

*Al Instituto Politécnico Nacional y
al Centro de Investigación en Computación
por brindar un ambiente propicio
para la investigación y el conocimiento.*

*Al Consejo Nacional de Ciencia y Tecnología
por facilitar enormemente, mediante sus apoyos,
la realización de este posgrado.*

“Tú, que me lees ¿estás seguro de entender mi lenguaje?”

(Jorge Luis Borges, Ficciones)

“another story leave it dark no the same story
not two stories leave it dark all the same like the rest
a little darker a few words all the same a few old words”

(Samuel Beckett, How it is)

“A civilização consiste em dar a qualquer coisa
um nome que lhe não compete,
e depois sonhar sobre o resultado.”

(Fernando Pessoa, Livro do desassossego)

“Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache.”

(Ludwig Wittgenstein, Philosophische Untersuchungen)

TABLE LIST

Table 1: Types of Named Entites.....	40
Table 2: Examples of possible phrases that contain the word “woman”	48
Table 3: Examples of NPVs that contain the expression “white-haired woman”	49
Table 4: Comparison between Verb Phrases and VPPs.....	50
Table 5: VPPs containing the word “stalked”.....	50
Table 6: PPPs containing the word “beneath”	51
Table 7: Examples of syntactic subordinates found within sentences	52
Table 8: Examples of Agent Subordinates.....	53
Table 9: Examples of Determined Subordinates	53
Table 10: Examples of WH-subordinates	54
Table 11: Examples of Complex Subordinates.....	54
Table 12: Descriptive phrases containing the word “dog”.	55
Table 13: Examples of extracted special clauses.....	56
Table 14: Ranking algorithm rules for NPV literary structures.....	65
Table 15: Ranking algorithm rules for VPP literary structures.	66
Table 16: Ranking algorithm rules for PPP literary structures.....	67
Table 17: Ranking algorithm rules for PHR literary structures.....	67
Table 18: The five possible starting sentences of the stories.....	76
Table 19: An example of algorithm generated sentences ready for macroplanning.....	80
Table 20: Subject Post-processing step.....	81
Table 21: NER substitution in the Post-processing module	83

Table 22: Macroplanning Structure	85
Table 23: Number of literary structures found in the experimental corpus.....	87
Table 24: The texts that were presented to human evaluators	88
Table 25: Aspects and concepts considered on the proposed survey	90

LIST OF FIGURES

Figure 1: An example of an automatic painting generated by AARON.....	13
Figure 2: Fabula model for story generation. The arrows represent possible causal relationships between elements of the fabula	15
Figure 3: An example of a story grammar taken from Thorndyke (1977)	17
Figure 4: Classical NLG System Architecture.....	24
Figure 5: The Classic Story Diagram.....	28
Figure 6: A comparison between the Classic Story Diagram and NLG Concepts.	29
Figure 7: Flowchart of the Knowledge Base construction process.....	44
Figure 8: Flowchart of the story generation process.....	46
Figure 9: VPPs are obtained from the syntactic Verb Phrases found in parsed sentences.	49
Figure 10: The constituent tree of a 25-word sentence.....	52
Figure 11: A graph visualization of the knowledge base showing NPV, VPP and PPP connections	60
Figure 12: Novel sentences are concatenations of intersected literary structures	61
Figure 13: Examples of NPVs connected to the node word <i>woman</i>	62
Figure 14: Examples of VPPs connected to the node word <i>say</i>	62
Figure 15: Examples of PPPs connected to the node word <i>to</i>	63
Figure 16: Steps that the algorithm follows to construct a story	69
Figure 17: Post-processing module steps at the sentence level.	79
Figure 18: The slider allows to show a tendency towards a chosen concept.....	89
Figure 19: Each question has five different parameters that are separately evaluated	89

Figure 20: Distribution of population by degree level.....	95
Figure 21: Distribution among evaluators' fields of study	96
Figure 22: Distribution of evaluators' reading habits	96
Figure 23: Positive votes received for positive concepts of each aspect.	97
Figure 24: Votes received for the negative concepts of each aspect.	98
Figure 25: Percentage of votes obtained in the creativity aspect of texts.....	99
Figure 26: Total semantic differential based on the total population responses.....	100

CONTENT

1.	INTRODUCTION	5
1.1	THESIS HYPOTHESIS	6
1.2	MAIN OBJECTIVE.....	7
1.3	SPECIFIC OBJECTIVES	7
1.4	CONTRIBUTIONS.....	7
1.5	THESIS OVERVIEW	9
2.	RELATED WORK	11
2.1	COMPUTER SYSTEMS ON CREATIVITY	12
2.2	TEXT GENERATION TECHNIQUES	14
2.2.1	<i>Problem Solving</i>	14
2.2.2	<i>Story Grammars</i>	16
2.2.3	<i>Corpus-Based Generation</i>	18
3.	THEORETICAL FRAMEWORK	21
3.1	LANGUAGE LEVELS.....	21
3.1.1	<i>Phonologic Level</i>	21
3.1.2	<i>Morphologic Level</i>	21
3.1.3	<i>Syntactic Level</i>	21
3.1.4	<i>Semantic Level</i>	22
3.1.5	<i>Pragmatic Level</i>	22
3.1.6	<i>Discourse Level</i>	22
3.2	CLASSIC NATURAL LANGUAGE GENERATION.....	23
3.2.1	<i>NLG Systems Architecture</i>	24
3.2.2	<i>Macroplanning</i>	24
3.2.3	<i>Microplanning</i>	26
3.2.4	<i>Surface Realization</i>	27
3.3	ABOUT SHORT STORIES.....	28
3.3.1	<i>What is a Story?</i>	28
3.3.2	<i>What is Creativity?</i>	29
3.3.3	<i>Contemporary Storytelling</i>	31
3.4	A MODERN NOTION OF MEANING.....	34
3.4.1	<i>Language Games</i>	34
3.4.2	<i>Possible Worlds</i>	35
3.5	NATURAL LANGUAGE PROCESSING TOOLS	36
3.5.1	<i>Syntactic Structures</i>	36
3.5.2	<i>Stanford Parser</i>	37

3.5.3	WordNet	38
3.5.4	JCN Similarity Measure	39
3.5.5	Named Entity Recognition (NER)	40
3.5.6	Stylistics	40
4.	PROPOSED ARCHITECTURE	43
4.1	LITERARY STRUCTURES EXTRACTION	47
4.1.1	Noun Phrases with Verb (NPV)	47
4.1.2	Verb Phrases with Preposition (VPP)	50
4.1.3	Previous Prepositional Phrases (PPP).....	51
4.1.4	Simple Phrases (PHR).....	51
4.1.5	Special Clauses (CLS).....	55
4.1.6	Substitution of Entities on Literary Structures	56
4.2	KNOWLEDGE BASE IMPLEMENTATION	57
4.2.1	Transitive and Auxiliary Verbs	57
4.2.2	English Prepositions	58
4.2.3	English Nouns and Pronouns	58
4.2.4	Proper Names	59
4.3	KNOWLEDGE BASE NAVIGATION	59
4.3.1	Sentence Construction	60
4.3.2	Word-Clause Similarity	63
4.4	PHRASE RANKING ALGORITHM	64
4.4.1	NPV Ranking	65
4.4.2	VPP Ranking.....	66
4.4.3	PPP Ranking.....	66
4.4.4	PHR Ranking	67
4.4.5	Roulette Wheel Selection	68
4.5	STORY GENERATION	68
4.5.1	Main Sentence	69
4.5.2	Handling Characters and Circumstances	72
4.5.3	Story Outcome	74
4.5.4	Introduction Sentence.....	75
4.5.5	Characters Description.....	76
4.5.6	Outcome Premises	78
4.5.7	Sentence Post-processing	79
4.5.8	Final Story	84

5.	EVALUATION AND RESULTS	87
5.1	CORPUS.....	87
5.2	STORY SELECTION	88
5.3	EVALUATION SURVEY	88
5.3.1	<i>Survey Design</i>	88
5.3.2	<i>Chosen Texts</i>	90
5.3.3	<i>Evaluators</i>	95
5.4	RESULTS	97
6.	CONCLUSIONS.....	101
6.1	CONTRIBUTIONS.....	101
6.2	FUTURE WORK	102
	REFERENCES	104

1. Introduction

Since computers exist there has been a debate between the capabilities of the mind and the machine. Is one better than the other? Are they equivalent models? Are they capable of solving the same tasks? There are strong arguments on both sides, and it seems that the problem will not be solved soon. Creativity is one of the arguments that some philosophers and psychologists use as a proof of what computers cannot achieve; however, these arguments are based on a misconception of what both intelligence and creativity mean.

One of the most tied concepts to creativity is storytelling. Stories exist since mankind has the ability to write. Knowledge, beliefs and fiction have jointly evolved, since written language (and storytelling especially) has been one of the preferred ways for humans to transmit information through generations.

On the other hand, automatic text generation emerged almost at the same time as computer science. Alan Turing designed, in one of his most famous papers, a simple test called the imitation game, which according to him could demonstrate if a computer is as powerful as a human brain (Turing, 1950). This famous test required the computer to automatically generate linguistic responses in order to deceive a human interrogator.

Huge amounts of works have been done since the Turing Test was proposed, and it is still a matter of debate if the test has been passed or not. What is still sure is the fact that a universal technique to generate human language has not been found yet. Furthermore, the current state of the art focuses more on practical and delimited applications that solve particular text generation problems, leaving creativity on the side. These perspectives have favored the conviction that artistic production cannot be achieved in computers.

This thesis provides arguments supporting that creativity can be emulated through computer programs. We chose automatic storytelling as an approach to demonstrate that the text generation task can go further than being a mere problem solving and content planning task.

1.1 Thesis Hypothesis

We believe that there is a simplified common method in the creative writing process that can be automatized. It is true that several frameworks approaching the problem of text generation already exist, but they approach it as a general task. We think that treating storytelling as a special instance of text generation can exploit the creative properties that language holds inside a corpus of fiction texts. Thus, we expect to find a method that is capable of using linguistic resources to produce fiction texts in a manner that can be called creative.

The assumption of creativity presents a major problem: Complexity. Even if we consider creativity just as a product of novel ways of achieving a goal, the number of combinations found when dealing with the ‘real world’ is astronomically huge. Recalling an example from literature, there is a short story called *The Library of Babel* (Borges, 1944), where the author takes us to a library that contains any possible book that could be written in the history of humanity. This library consisted of 410 pages long books (larger books would be divided in volumes), with 40 lines of 80 characters, with an alphabet of 26 symbols; the content of the first book would be just blank spaces in all pages of the book, and going through every coherent book (for example the book of *Fictions* from Borges himself) all the way to the last book which would contain just z’s (Borges limited his library to just Latin alphabet written books, but even with this significant reduction the number of combinations is gigantic). There would be even more books in this library than particles in the known universe, making this library physically impossible (Dennett, 2013). If this imaginary library was organized randomly, there is no need to mention that searching for a specific book can be considered an NP-Hard problem.

The metaphor of Borges reveals the combinatory problem that emerges if a brute force algorithm is designed to generate texts. Also by following this metaphor, we can deduce that the existence of a novel text only depends on a different combination of the same existent resources. According to our hypothesis, our proposal is a heuristic that uses simple syntactic

and semantic properties found in a text corpus in order to generate novel and coherent fiction texts based on what has been already written.

1.2 Main Objective

The main objective of this thesis is to generate texts with coherent novel sentences, by learning and adapting them from previous fiction texts.

1.3 Specific Objectives

There are a few number of steps that need to be taken in order to achieve our main objective. First, we need to construct a corpus of fiction texts in order to extract relevant word relations that exist on books. Next, we have to detect syntactic patterns within sentences and isolate them in order to use them as the main building blocks for novel sentences.

Once we have isolated word relations and syntactic patterns, the task is to generate new sentences that hold internal syntactic and semantic coherence. And finally, the last step is to find an assembling procedure to join the phrases into a text that can be called a story.

1.4 Contributions

The state of the art has many examples of storytelling generation engines, some of them will be listed on chapter 2. They tend to follow three main approaches: problem solving, story grammars, and corpus-based. The problem solving approach sees story generation as a search task, where a solution needs to be found from a current state (the story beginning) until a goal state (a logical story outcome). Story grammars work better at the language production level, although its problem is an excess of rigidity on the output language, and also it relies on previously constructed databases. Lastly, the corpus-based approach offers more flexibility since it can infer linguistic knowledge from statistics, but it has many problems at constructing surface language.

In addition, many of the current works use a general architecture, which divides the generation problem in three layers: Macroplanning (document level planning), Microplanning (sentence level planning) and Surface Realization (the actual language that is presented as an output).

This architecture is the most accepted approach; however, since it is a general solution, it tends to produce a monotonous language that lacks of style, which is a main characteristic of literary texts.

We found that all these works already propose formalized ways to construct texts that can be called stories. Most of them function well from the logical point of view, at story planning and character constructions, but they do not give the necessary importance to linguistic production. Moreover, since they formalize the story generation, they do not focus on the aspect of creativity, which we consider to be an intrinsic property of any good story.

What we propose is to generate stories that give the sensation of literary style instead of just a carefully planned and structured text. We intend to give more importance on the linguistic layer of stories. Each text genre has its own syntax and vocabulary, thus instead of leaving surface realization to a general module, we will learn sentence style directly from our corpus. We know that manner itself gives clues to interpretation, therefore, this work focus on producing stories that mimic the manner of literary texts. Finally, we propose a method to coherently combine phrases in original ways to form new sentences, producing the impression of creativity.

Our framework consists of two steps: the knowledge base construction and the generation engine. The knowledge base contains word relations and syntactic patterns, which are directly extracted from the parsed corpus. Once we have a defined knowledge base, the generation engine navigates through the base to obtain and properly rank the desired phrases that will be used to construct the novel sentences. Then, the generation engine handles how to ensemble sentences to construct the desired story. And at last, it outputs the story in a proper fashion.

Dividing the framework in two steps, instead of searching for alternatives while generating, allows us to have a bigger knowledge base without compromising the processing time at the generation step; thus, we consider the speed of our algorithm as another important contribution.

1.5 Thesis Overview

This thesis consists of six chapters, which will be explained next.

Chapter 2 starts with a review of the existent related works. It is divided in two main sections: First, we list the works that intend to automatize creativity, they are not necessarily about text generation, but they provide important concepts that we want to follow on our own work; the second section explains text generation techniques in general. As we explained, there is not a specialized story generation framework, and this is one of the main weak points that we wish to fix by combining some of the existent techniques.

In Chapter 3 we introduce all the concepts that will be used to construct our framework. We present concepts taken from philosophy, linguistics and computer science that will help us to propose our story generation framework. We speak about the different levels in the study of language and how each level contributes to meaning in texts. Then we describe how the classical natural language generation architecture works. Next, we briefly speak about the theory of short stories and creativity, together with a philosophical approach to meaning in language. Finally, we list some of the language processing tools that will help us to process our corpus and extract the desired patterns, which will serve as building blocks on the generation step.

Chapter 4 explains how do we combine the listed concepts from the previous chapter in order to fix some of the problems that we noted on the existent works. Here we present the five main syntactic patterns that will be extracted from our corpus. Afterward, we describe the implementation of our knowledge base, how it is constructed and how it can be navigated to obtain and ensemble the desired full sentences that will be used in the story. We also list the default knowledge that is stored in the base, which will help the algorithm to build coherent sentences. Next, we describe the generation module; we list which sentences need to be put together to complete a new story, as well as the criteria used to select them. Lastly, the post-processing module is described, where all sentences are checked for spelling and grammatical coherence in order to display the proper output.

In chapter 5 we present a method that evaluates the quality of the generated stories. We describe how we designed the survey we used to analyze the aspects that we consider

important in storytelling production: coherence, interest, syntax and creativity. Finally, we present the results of our experiments.

Chapter 6 lists the most important conclusions that we extracted from our work. We also point out some of the problems that arose while evaluating and at the end we propose a few aspects for future work.

2. Related Work

There have been many different approaches to both creativity and automatic story generation. The problem of creativity exists as an abstract problem since the beginning of computer science, but the capabilities of computers as producers of creativity have been hardly explored. On the other hand, automatic story generation has been deeply studied but only as a particular instance of the text generation problem. Because of this, the available techniques tend to focus only on the formalization of storytelling in order to fit into previous text generation frameworks. We can distinguish three main approaches to automatic story generation:

- **Problem solving approach:** sees storytelling mainly as a plot generation problem, and tries to adapt Artificial Intelligence techniques to fulfill this task.
- **Story Grammars:** tries to apply the nature of production grammars into the storytelling area, designing grammars that produce text with the structure of stories.
- **Corpus-based Approach:** looks for less dependency on human intervention when generating stories. Tries to benefit from knowledge present in previously written texts.

In this chapter, first we will analyze some of the works that have pursued creativity, from small text generators to artificial painters. Then we will describe the current approaches to text generation, some of them are not intended as story generators, but they contribute in some way for explaining our proposal on chapter 4.

2.1 Computer Systems on Creativity

Since the 1950's (almost at the same time as formal computing emerged), different programs have been made in pursuit of creativity. They tend to be small and simple scripts that do not directly model creativity, but surely inquire about what can be done with it. Most of these works involved natural language, as written language was the straightforward approach to follow with a machine that process symbols. In (Montfort & Fedorova, 2012) we can see a survey about the most important small-scale works that have been made in order to pursuit creativity as a computational activity.

The survey begins in 1952, when Christopher Strachey created a "Love Letter Generator", which is considered as the first experiment in digital literature. It ran on the Mark I, producing texts such as "YOU ARE MY EROTIC APPETITE: MY SWEET ENTHUSIASM. MY LOVE FONDLY WOOS YOUR CURIOUS TENDERNESS, YOU ARE MY WISTFUL SYMPATHY". Its purpose was to parody the repetitive process that writing a love letter meant, but it also demonstrated how examples of human writing could be emulated, in this case with just a set of strings, some templates and a random combination of them.

Several works followed this first attempt of emulating literature. In 1959, Theo Lutz generated stochastic texts based on Kafka's "The Castle", producing different *kafkian sentences* that were not previously present on the book. In 1998, Nannette Wylde's "About so Many Things" produced pairs of sentences, the first one starting with a "He" and the second one with a "She". Wylde claimed that this showed how perception of sentences changes according to the gender of its subject. Nick Montfort's "Through the Park" takes a story as an input and choose which stanzas to erase in order to give the output text an appearance of ellipsis, forcing the reader to use her imagination to fill the story in. All these works are now considered digital literature, maybe even digital creativity, and as we shall see on the next chapter there are several reasons that support this claim.

A very different approach to creativity was reached through painting, also with a much larger scale program: AARON. This program was created in the seventies by the painter Harold Cohen. He began experimenting with programing in 1968 at UC San Diego, and decided that an automatic painter program could work by following a set of rules, which he himself used

while painting (Cohen, 2010). In the first stages, the computer produced an output on the screen and Harold drew the sketches following exactly the program output. At later stages, the computer was plugged to a plotter, and also the source code was expanded, so the paintings were produced automatically, and novel paintings appeared over time since the program used its own output as an input for the next piece. In fact, AARON is still active today, and its creator claims that the code, written in LISP, has evolved a lot since its first version and that he no longer controls what AARON is going to do next.

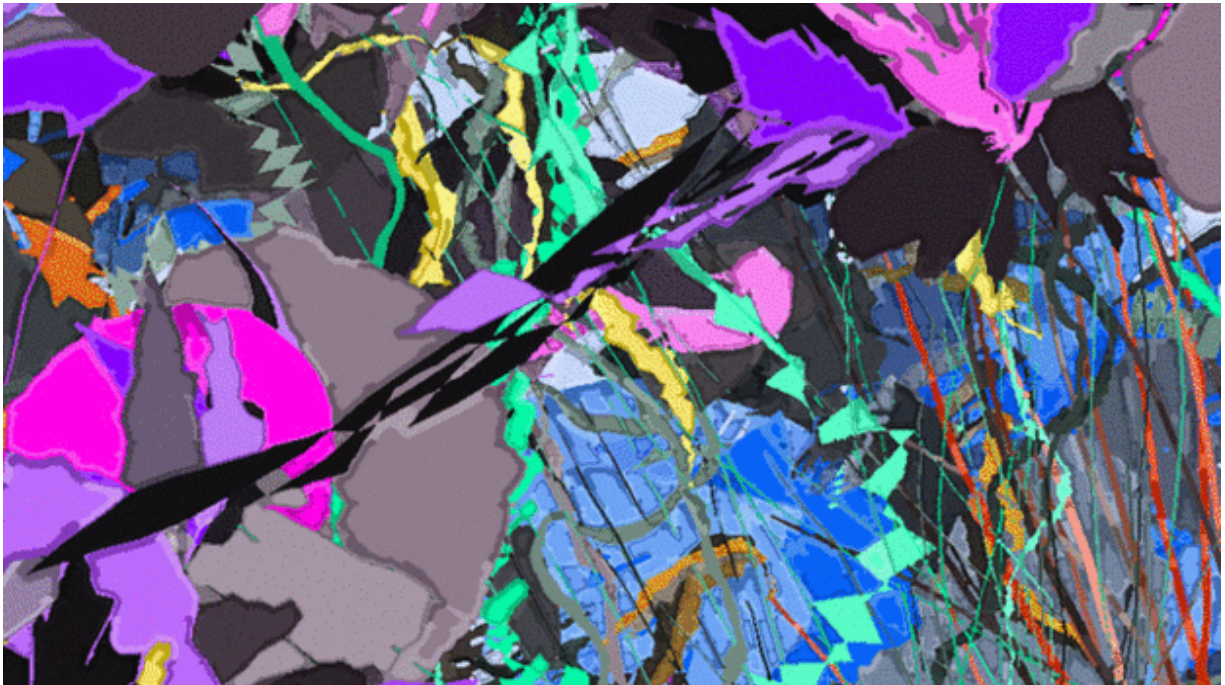


Figure 1: An example of an automatic painting generated by AARON

On 2010, Harold Cohen mentioned that he considered his program to be creative, since the output that it produces is not an explicit intention of the programmer (he just gave the instructions of what was valid and what was not) and of course it is not only a product of computer intentionality (since the source code has been made by the programmer), but it is the interaction between programmer, rules and outputs what has created a novel way in which painting is done, and that is for sure what creativity means.

A more recent work on creativity was made in the pursuit of generating song lyrics that hold a specific style (Barbieri, Pachet, Roy, & Esposti, 2012). This approach uses Markov processes and constraints to approximate to certain given styles. In this work, a semi-automatic generation of lyrics is made following the style of Bob Dylan and satisfying the structural constraints of the song *Yesterday* by the Beatles. By imposing rhythmic and Part of Speech (POS) templates, and the semantic relatedness by the Wikipedia Link-Based Measure as the Markov constraints, the algorithm achieves to relate the output to the Beatles' song; and doing a statistical analysis of Bob Dylan's lyrics it manages to imitate the content of the song. It is important to emphasize that it was a semi-automatic work, where the system only proposed generated verses, and it was a human who made the decisions of which verses fit better to the final song that was produced.

A direct mixture of creativity and storytelling in a big project did not happen until MINSTRELL (Turner, 1994). This system, following a problem solving approach, intends to generate stories as a combination of character and authorial goals. It generates stories about King Arthur, but its particular attention was on making these stories creative. The system uses case-based reasoning, and it remembers previous solutions in order to avoid them on the next iterations; and as a consequence, encourage creativity (diversity) on every generated text. It is also able to adapt previous solutions to new problems, so its creativity also relies in the way it solves the presented problems.

2.2 Text Generation Techniques

2.2.1 Problem Solving

A problem solving approach means exploring a problem space to try to find some path from current state to goal state. Seeing story generation this way is to have an initial situation as an input and try to develop a logical way to reach a story ending.

One of the first works with this approach was TALE-SPIN (Meehan, 1977). It generated stories about animal characters in a fashion similar to Aesop's fables. Each story was already supplied with a set of goals and rules that were searched in order to advance the plot and fulfill a desired goal. This work had already made knowledge bases that included the possible

actions every character was capable of, as well as the relationships and personalities. Text production then was an exploration of that search space and the application of the set of given rules.

A more recent approach follows the same idea but also presents a whole fabula model that every story should follow to accomplish text production (Swartjes & Theune, 2006). Their work does not perform text generation, but proposes a whole model that is inspired in Emergent Narrative, which is the representation of stories through event sequences that can then be fed to a story generation engine. A General Transition Network is proposed, based on story comprehension studies, which is composed of six elements: Setting, Event, Internal Response, Goal, Attempt and Outcome. Besides the elements there are four types of causalities that can link the elements: Physical Causality (something happens that changes the story world), Motivation (a character wants to accomplish something), Psychological Causality (characters beliefs) and Enablement (because an event A happens then an event B can be possible).

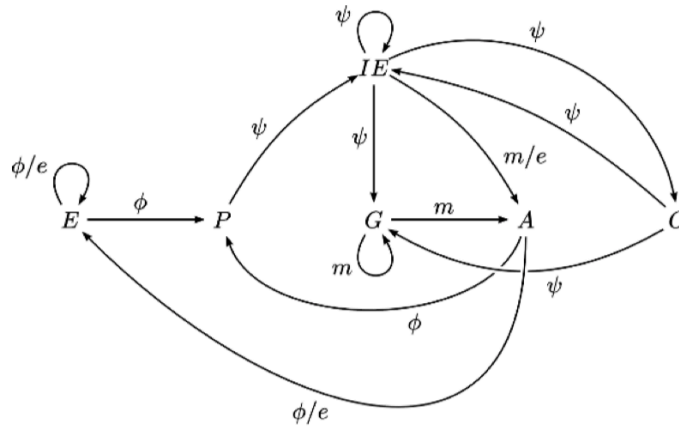


Figure 2: Fabula model for story generation. The arrows represent possible causal relationships between elements of the fabula

This is a very high level approach to story generation, and despite it looks like a fairly simple and logical approach, it does not clarify how to automatically map texts into the mentioned elements and transitions. On the other hand, supposing that a clean extraction of the elements is done, there is still a big gap between the proposed narrative structure and the actual linguistic production. The work is only done as a theoretical approach as was not continued.

A very different approach was made in MEXICA (Pérez y Pérez, 1999). This is a system that tries to define and then emulate the *Engagement and Reflection* process in which a human author gets involved while creating stories. It is a relevant work in terms of introducing concepts to text generation beyond the mere problem-solving task. However, it relies strongly on user input because it needs a whole set of previously written stories to learn from it and then transform it into a single novel story.

According to this work, the Engaged State can be described as a state in which the writer is intensely involved in the production of material related to the task, and such a production is guided by the author's own cultural constraints. The Reflective State can be described as a state where the writer evaluates the current work and deliberately explores and transforms it. Having these two concepts in mind Pérez y Pérez realized that they could be directly mapped into a genetic algorithm framework, where a population of stories that could evolve through generations and, with a proper fitness function, the *best* story could be determined, just as a human author would do.

2.2.2 Story Grammars

A grammar is a set of production rules that rewrite strings of symbols until reaching terminal nodes. This was one of the first approaches followed in the pursuit of text generation. Several works have been created in this area; however, they tend to focus more on structure than content, creating rigid and cognitively poor stories (Black & Wilensky, 1979). Another arguable characteristic of grammars is their insufficiency to capture stylistic literary devices, proving that story grammars can be helpful in many aspects but should not be used as the only generation device.

- (1) STORY → SETTING + THEME + PLOT + RESOLUTION
- (2) SETTING → CHARACTERS + LOCATION + TIME
- (3) THEME → (EVENT)* + GOAL
- (4) PLOT → EPISODE*
- (5) EPISODE → SUBGOAL + ATTEMPT* + OUTCOME
- (6) ATTEMPT → EVENT* | EPISODE
- (7) OUTCOME → EVENT* | STATE
- (8) RESOLUTION → EVENT | STATE
- (9) SUBGOAL | GOAL → DESIRED STATE
- (10) CHARACTERS | LOCATION | TIME → STATE

Figure 3: An example of a story grammar taken from Thorndyke (1977)

To overcome some of the problems listed above, works that follow this technique turned into literary theory to enrich the power of their grammars. One of the most cited authors when trying to formalize the storytelling process is Vladimir Propp. He made an analysis of Russian Folk Tales and defined a finite set of roles (such as hero, villain, donor) and functions (e.g. the hero's motivation, the battle between hero and villain) that every character in those stories followed (Propp, 1968). The rigor of this work makes it very suitable for automation (Lakoff, 1972). The work of Lakoff was strongly followed in (Gervás, 2013) where the best generalization of Propp's functions was made, turning Propp's morphology into a formal grammar. Gervás created a system that generates Russian folktale-like texts using Propp's roles and functions as a formal grammar.

In this work the story generation is developed in three steps:

- Plot Driver generator: employs an algorithmic procedure for generating a sequence of character functions considered valid for a tale.
- Fabula generator: given a valid sequence of functions, progressively selects instantiations of these character functions in terms of story actions.
- Flow generator: given a fabula where all variables have been replaced by constants, produces a flow for the story.

This approach proved to be highly effective, however its biggest fault, as in other systems that follow the formal grammar approach, is scalability in terms of text diversity. Since Propp's roles and functions were designed based on Russian folktales, a grammar based on his work can just be applied to generate stories with the same characteristics; therefore, stories generated based on Propp's grammar will always resemble a Russian folktale. Many interactive storytelling works used the grammar made by Gervás as a baseline because of its formality, but if the aim is to generate creative texts, then a different technique needs to be applied.

2.2.3 Corpus-Based Generation

In general, text generation software is intended for a very well defined area. This means that certain words, expressions and text content are expected depending on the area where they will be used. For example, if someone wants to create an automatic weather report system, it will only need to use words related to weather reports, and its general structure only needs to resemble that of a human weather report, which does not strongly differ among weather reporters. In (Reiter & Williams, *Generating Texts in Different Styles*, 2010) text generation systems are studied to stand out the importance of style in the output. Although these systems are not in the storytelling area, Reiter points to some stylistic aspects that can be taken into account, even in technical texts, to avoid a rigid text generation, all these with the help of corpus analysis.

The first aspect is explicit stylistic control: to give the user the control over some parameters like sentence length, verbosity mode, aspects that should be emphasized, etcetera. Another aspect is to conform to the specific genre in which the application is developed, studying concrete corpora (if the system is a weather report generator then only weather reports should be studied) to analyze the common vocabulary that humans use to communicate certain ideas. Finally, an individual author from the same area can be studied in order to imitate her communication techniques, vocabulary and syntax in order to make decisions based on what is found. These aspects can be managed by developing rules and/or using machine learning techniques to classify the important features of texts.

One of the main examples that is mentioned in this work is SKILLSUM. It was developed by Aberdeen University to generate feedback reports for people who have just taken an on-line assessment. Its aim is to inform people about the specific problems, if any, that were found in their assessment and to give some advice to improve those basic skills. This system uses the basic Natural Language Generation Architecture (which will be explained on chapter 3) to produce the final text output. What is important about it is its use of previous psycholinguistic knowledge extracted from similar reports made by humans. By doing so, the actual text that will conform the output should hold a style that resembles human reports.

The most relevant contribution from Reiter is the emphasis that he made on style, contrasting previous works, which focus more on generation as an AI task than on the actual language that will be displayed to the user. Reiter also points out several issues that arise when working with this approach, but the results that can be obtained with it are very interesting.

In the specific area of automatic corpus-based storytelling, one of the most relevant works is found in (McIntyre & Lapata, 2009). They propose a complete data-driven approach to story generation that does not require extensive manual involvement. Their focus is mainly on children stories, using a corpus made from a specific selection of them. They recognize two phases in the story generation systems: the creation of a plot and the transformation of the plot into actual text. By doing so, they mainly focus on extracting the plot from predicate-argument, predicate-predicate co-occurrence statistics gathered from corpora, and leave the actual language production to a standard language model that is already provided by RealPro (Lavoie & Rambow, 1997), which is a text generation engine that performs syntactic realization output based on abstract syntactic specifications.

The importance of this work lies in the high value it gives to data, giving a lot more flexibility than the problem solving and grammar approach. This is mainly because the content of the output directly depends on the corpus that is feeding it, and the plot is constructed based on the statistic findings from the corpus, avoiding the necessity of creating explicit rules in the stage of planning. However, as they only focus on plot creation, the language that is actually produced results in a sequence of simple sentences that often are monotonous and inexpressive, and the output loses all the specific style from input corpora.

In corpus linguistics the field of compositional semantics is well known for the search of mapping meaning into sentences or utterances longer than individual words; this is done by transforming utterances into vectors that can be mathematically manipulated. In “How to make words with vectors” (Dinu & Baroni, 2014) intend to revert this technique in order to generate text based on given vectors. This is a complete mathematical approach, often used for paraphrasing or machine translation, which transforms vectors through composition operators to obtain semantically similar expressions. In this particular work it is used, among other experiments, to transform one word into two-word noun phrase (i.e. “reasoning” maps into “deductive thinking”; “folk” maps into “local music”).

Unfortunately, this work is aimed to tasks other than story generation, so its efficiency can only be seen in the fields of machine translation and paraphrasing; but it is relevant to this thesis because of considering generation as a mapping problem (from existent meanings in corpora into novel text), instead of just being a planning problem.

We have listed some of the most important works on creativity and story generation, including a brief explanation of their general objectives and also some of the problems they present. As noted above, works that focus on creativity tend to attack a theoretical piece of the generation problem, leaving many steps unfinished, or strongly relying on human input to complete them. In contrast, automatic storytelling tends to be a more formal approach, but with its formality it loses some of the *properties* that human creations tend to have such as style, syntactic dynamism, vocabulary diversity, and creative situations.

In the following chapter we list different concepts that will help us to surpass these problems. Some of them seem to be far away from each other, but in chapter 4, where we explain our proposal, we show how are they combined into a single framework that is able to automatically generate fiction texts that hold semantic coherence and a sense of creativity and style.

3. Theoretical Framework

3.1 Language Levels

Natural language is the main communication tool that we use to transfer information to one another. We use it to organize our knowledge, and to try to comprehend reality by segmenting it in a standard way, so we can be able to understand each other. Language can be studied mainly in six levels (Calvo, 2013):

3.1.1 Phonologic Level

Phonetics is the study of language through sounds. It focuses on word pronunciations, syllable division and word stressing. It also studies the acoustic properties and the human perception of them.

3.1.2 Morphologic Level

This level focuses on the study of morphemes. A morpheme is the basic unit of written language. It includes roots (when a morpheme is equal to a word), suffixes and prefixes, declinations, gender and number.

3.1.3 Syntactic Level

It studies the function of each word within a sentence, and the relationship that words have among each other. There are two main models that study syntax:

- **Dependency Model:** it analyzes sentences in a tree fashion, where the root is the main word of the entire sentence, and each leaf represents a modifier of its parent inside the tree.
- **Constituent Model:** A constituent can be a word or a group of words (also known as chunks) that represent a single syntax unit. This is a hierarchical model where constituents are broken recursively into simpler chunks or sub-constituents, the process is repeated until each constituent is an individual word.

3.1.4 Semantic Level

Semantics refers to the understanding of the sense of the sentence as a whole. It includes the sense of each individual word, but this is not enough, because words often have more than one sense and, in order to get the correct sense of a word, the context has to be analyzed including the syntactic role that the word plays in the specific sentence. Also at this level coreference resolution inside sentences is studied, which is the correct use of articles, quantifiers, and pronouns referring to previously mentioned concepts. Some of the most commonly used tools that help in the study of semantics with a classical approach are ontologies, thesauri, semantic networks, and frames.

3.1.5 Pragmatic Level

At this level, intentions that rely on the text are studied. Pragmatics also depends on context, but it goes beyond semantics in the sense that it studies implications that are not explicitly present in the text, such as writer's (or speaker's) intention, manner and previous knowledge about content; time and place where the utterances were made, etcetera.

3.1.6 Discourse Level

Finally, at this level, relationships among sentences are analyzed. Any given piece of text communicates an idea as a whole; this means that sentences are not isolated entities but parts of a bigger concept that needs to be transmitted. Discourse level analysis includes co-reference resolution among sentences through the entire text.

3.2 Classic Natural Language Generation

Natural Language Generation (NLG) is the process of constructing natural language outputs from non-linguistic inputs: its task is to map meaning to text. It is the opposite task of Natural Language Understanding (NLU), which is the mapping from a given text to its intended meaning (Jurafsky & Martin, 2000). While the field of NLU studies the aspects of transforming human language input into suitable data structures for the machine, NLG focuses on using language knowledge in order to produce outputs of information that are as closest to human utterances as possible.

At the beginning, scientists focused on NLU because it appeared to be a much more difficult problem than NLG (Reiter & Dale, Building Natural Language Generation Systems, 2000). This was because NLU faces the problem of controlling the language complexity of its input; however, while it is true that NLG can control the level of complexity when producing language, it faces a list of many different problems that are not present in NLU. The main problems that NLG tries to solve are both at sentence and document level. Some of the questions that arise are:

- How to communicate a specific idea?
- How to choose the lexical output?
- Which is the best way to express a concept?
- How to remain coherent through the entire text?

Nowadays NLG appears as an attractive field of study since it has a lot of practical applications. Because it is a complex task, normally NLG systems focus on the specific applications to which they are aimed, and their architecture and linguistic resources are designed *ad hoc* to fulfill those specific tasks for which they are created. Nonetheless, a fairly common architecture for building NLG systems can be described (Reiter & Dale, Building Natural Language Generation Systems, 2000). We have called this the Classic Natural Language Generation approach.

3.2.1 NLG Systems Architecture

There are four main aspects any NLG system should include to correctly communicate its intended message: quality (use reliable information sources), quantity (the amount of words that need to be said to complete communication), relation (order information correctly) and manner (the way it is said) (Grice, 1975).

Each of these aspects is addressed on a different layer of the NLG system. To apply these aspects is not as straightforward as it may seem, since the application of one aspect affects another one, and an optimal equilibrium among quantity, relation and manner is desired.

Every NLG system relies on a knowledge base that contains all the possible information that could or should eventually be communicated to its user. The Classic NLG architecture proposes a top-down hierarchical approach where, based on an initial goal, a general plan of the final document is done first (macroplanning); then the sentence specific design aspects (microplanning); and finally, the definitive linguistic production (Surface Realization), that is to say, the *actual* text that will appear as the output of the system, which is often referred as final document.

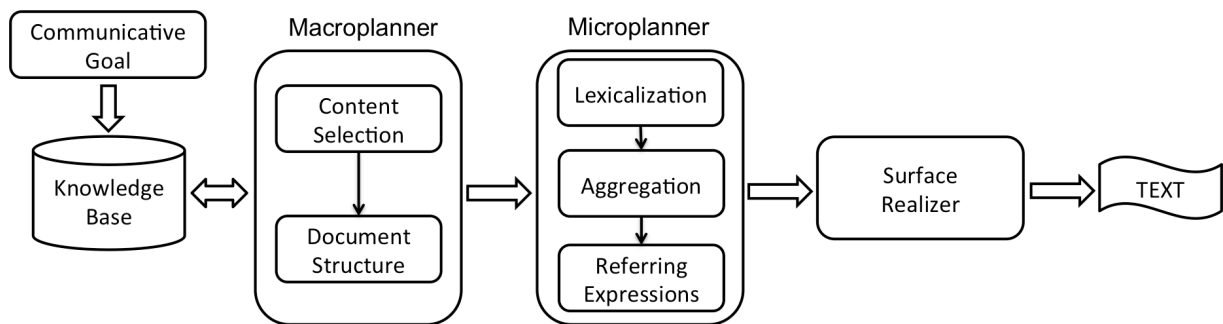


Figure 4: Classical NLG System Architecture

3.2.2 Macroplanning

The input at this level is the same as the input of the system: a specific goal that needs to be achieved. Based on this input, content is chosen from the knowledge base, and then the main skeleton of the document has to be created. The final output of the macroplanner is a

data structure that represents the document plan; such structure will be taken as the input of the microplanner.

This step focuses more on “what should be said” than on the manner in which it is said. This task is divided in two main sections: Content Selection and Document Structuring; that is, to retrieve from the knowledge base the correct information that fulfills the request (quality) and to arrange it in a proper way to communicate the desired idea successfully (relation).

3.2.2.1 Content Selection

Usually there is a huge amount of data stored in a knowledge base, and a method should be applied to identify the data that is relevant to achieve the desired goal. The simplest technique to do this is by defining rules about what information should be gathered in each request case; however, this is clearly a very limited approach if it is done manually (although, if the application is perfectly delimited this is still the more efficient way to work, since the concepts and relationships that will be presented in texts can be easily defined in ontologies or dictionaries (Reiter & Dale, Building Natural Language Generation Systems, 2000).) It should be mentioned that there have been attempts to automatically learn rules via stochastic methods and genetic algorithms, where relevant rules are inferred from corpus information. Another common approach is the use of already separated structured datasets that are consulted only on specific requests, so the content selection problem is avoided.

3.2.2.2 Document Structuring

Having the correct content to communicate is not enough. A document is not just a randomly ordered information piece. Usually basic discursive coherence is expected, including sequenced sentences; relevant information at the beginning, followed by the elaboration of that information; and a correct hierarchical order of ideas, that is, main ideas go first followed by secondary arguments.

As with content selection, it is not uncommon to have a pre-defined set of document structures or schemata that each generated piece of text follows. This is because the

structure of a document is highly genre-dependent, meaning that a user expects to see different elements on a text depending on the intended purpose of it: the structure of weather reports, short stories, medical charts, or financial reports are without question very different among each other.

3.2.3 Microplanning

The job of the microplanner is to complement the document structure giving it more directives at the sentence level. If the NLG application is simple enough, language can be generated directly from the document plan omitting this step. Nonetheless, having a microplanner gives more flexibility on the possible outputs of the system, since it allows a better control over vocabulary choices (Lexicalization), the length of the phrases (Aggregation) and, in general, the manner in which information will be communicated.

It is also important to say that this step still manipulates internal mark-up language with indicators and instructions. This information is added to the document structure and given to the surface realizer, which is the step where the natural language is actually produced.

3.2.3.1 Lexicalization

Lexicalization consists in choosing the particular words that need to be included in the sentence. In order to choose the correct words, the syntactic requirements of the sentence have to be considered. This is an important step, because there are several ways in which a piece of information can be expressed, and the grammatical form in which it is uttered sometimes affects the interpretation of that same piece of information. Also, the microplanner should be able to choose among words that have intersected meanings, and depending on the intent of the specific sentence, it should decide what is the correct term that will appear on that sentence.

3.2.3.2 Aggregation

It is also important to decide how much information needs to be communicated or omitted in each sentence. This is primarily done to be as informative as possible and at the same time select only relevant information. Furthermore, aggregation deals directly with style,

which is also a genre-dependent feature. For example, if the genre of the specific application demands only short sentences, then, the aggregation module can break up a big sentence from the document structure and transform it into two or three separated sentences; likewise, if there are several similar sized consecutive sentences they can be modified or merged, to avoid a repetitive or tedious reading of the text.

3.2.3.3 Referring Expressions

Any domain is populated by entities. When a large piece of text is generated, it is common to refer several times to that same entity. This is a semantic problem for the microplanner, because, in order to avoid a rigid reading of the text, that same entity should not always be referred directly but with the correct use of pronouns or quantifiers. For example, if we read the sentences: “At first, the woman was sleeping. When the woman woke up, the woman made breakfast.” we immediately assume there is something wrong with the text. We strongly prefer to read the following: “At first, the woman was sleeping. When she woke up, she made breakfast.” The microplanner needs to track these entities through the document structure and arrange the pertinent changes at sentence level to improve the flow of the text.

3.2.4 Surface Realization

Finally, in this step the linguistic production occurs. Based on the document plan (enriched by the microplanning annotations), actual sentences are produced. This is a phase where almost every NLG system differs since there are several techniques used to fulfill this task.

In order to produce natural language, lexical, syntax and grammatical aspects should be considered. There are several modules that already approach this step with various techniques such as subject meaning specifications, lexicalized case frames, canned text or corpus analysis. Surface realization itself is a big field of study, hence many of the works whose main goal is broader than mere surface realization (such as storytelling) concentrate only on macro and micro planning, focusing on the production of a required output for one of these existent modules in order to complete the actual text production.

3.3 About Short Stories

3.3.1 What is a Story?

In order to be able to generate short texts that can be called stories, we need to define what a story is. In *Poetics* Aristotle says that the same way as painters imitate life through color and form, and musicians through rhythm and harmony, also poets do this imitation “by means of language alone”. He also distinguishes three ways in which all arts in general perform this imitation: the medium, the objects, and the manner. In the specific case of written text, the medium would be words, the objects would be the characters and sceneries, and manner could be seen as language style or narration.

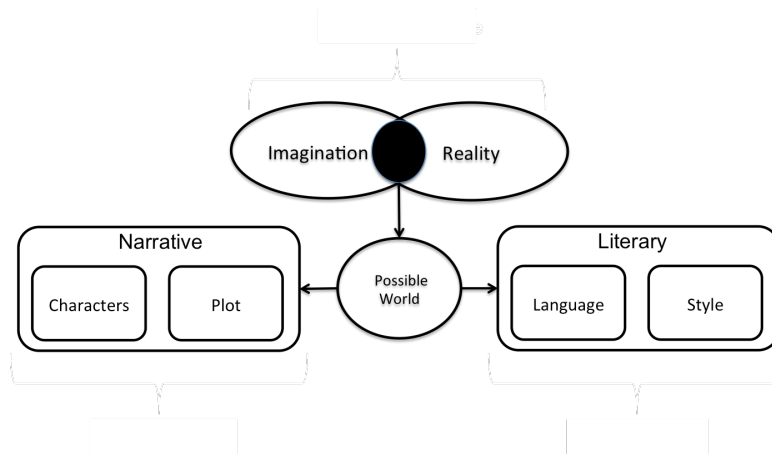


Figure 5: The Classic Story Diagram

Based on Aristotle, we can define a story as an intersection of imagination and reality, a *possible world* created by means of language (See figure 3). By possible world he means that anything can exist or happen inside the *world* where the story is developing, as long as it remains coherent to itself; meaning that once some *fact* is established inside the story, it cannot be contradicted.

In addition, we can distinguish two main levels in which every story can be analyzed: narrative and literary. The narrative level is the skeleton of a story, it is where the characters or agents that are able to make choices through the story are defined, including their capabilities and circumstances; and also where concatenation of events that occur to

our agents are put in a logical sequence called narrative arc, which controls the plot of the story (McKee, 1997). On the other hand, there is the literary level where the main focus is language and style: which words should be said and in what order should they be said, so we can communicate the intended meaning, the length of sentences, the flow of prose, etcetera.

On the whole, it needs to be said that an intersection between the classic story definition and the natural language generation architecture concepts can be directly found, as shown on figure 4; for this reason, we presume that both approaches can be combined to formally generate texts with literary style.

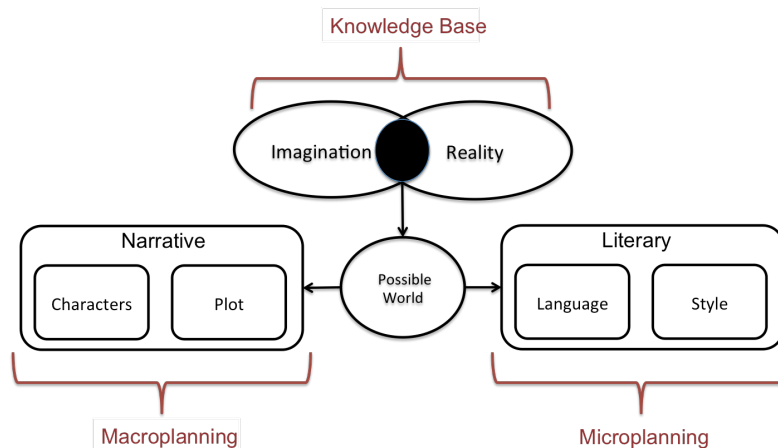


Figure 6: A comparison between the Classic Story Diagram and NLG Concepts.

3.3.2 What is Creativity?

For many years, creativity has been considered a “special” characteristic of the mind that is impossible to emulate through an algorithmic approach. As intelligence, creativity has a wide variety of definitions. Therefore, a definitive answer to what creativity is cannot be given. In former definitions creativity was considered a spiritual entity; a form of divine manifestation that occurred to some people who were merely the media to communicate these ideas. Presently, some people still have a huge stigma on creativity, considering it as a special gift that just selected groups possess, such as writers, painters and all kind of artists. Many times creativity has been considered as "completely breaking the rules" or “getting totally out of everything that is known today”, a once in a life time occurrence in

the universe, hence it is impossible to construct any creative mechanical device since we do not know when the next “creative spark” is going to occur.

It is true that through art we can presence the best manifestations of creativity; however, it is not true that merely a selected group of people possess creativity. On the contrary, creativity should be considered a general feature of the mind, an ability to generate and pursue any kind of project or *goal*. Although it is surely more developed in some individuals, creativity can be seen as the *process* of *searching* for useful links between concepts (*knowledge*) in order to achieve the desired goal (Marina, 1993). Normally, the creation of these links is not straight forward, in fact it is a difficult task, and creativity truly relies on the capacity of effectively finding these links: the novel a link is, the more creative the person who found it is considered.

As seen on the last paragraph, merely by defining creativity we already found an intersection with Artificial Intelligence (AI), since we have mentioned on the definition familiar concepts used everyday on AI such as search, process, knowledge and goals. On his paper “The Mechanics of Creativity” Roger Schank explains his reasons to believe that an algorithmic schema can be extracted from human creativity. He defines creativity as the capacity to bend pre-established rules and the ability to apply and modify different paradigms (from different fields of study for example) to solve a given task (Schank, 1992). Creativity then would be a novel chain of causation that leads from a given start point to the desired goal; the discovery of a new path through already known questions and answers can also be considered creative, especially if it is an “easier” path. He even goes one step forward and says that the mere exploration of the search space could be considered as creativity, especially when the search space is huge (as it usually happens), since a novel *strategy* needs to be developed in order to find something that wasn’t previously found.

Even humans are not always creating new ideas. In fact, most of the times humans rely on previously known solutions and their behavior tend to follow rules that were previously learned in society. It is only when we need a more effective solution, or when we face a totally new problem, that we begin the search for a different solution. Of course creativity is not only focused on problem solving, it manifests on any intentional desire that appears

on the mind. The important thing that Schank points out is that the search for novel paths between a desire and a goal occurs under a previously given knowledge base, and the *creative discovery* occurs through a novel combination of paths inside the given search space.

This approach to creativity does not solve everything though; it even generates more questions to answer. There is, for example, the fact of the enormous size of the search space, the problem of decision making through the paths that we encounter, and what may be the most difficult task: how do we know that the solution we found is better than the previously known? How do we ensure we already found the best possible solution? Creativity is also about solving these kinds of questions, and we can certainly claim that AI systems can engage with creativity more easily than it is usually considered.

3.3.3 Contemporary Storytelling

The 20th century brought big changes on how humanity sees the world. One of these changes was art, which developed several movements or schools to explore different ways of creating artistic artifacts. Literature was not the exception, and new linguistic experiments were made on storytelling and poetry. Stories were not anymore just written to show an anecdote, they were not mere logical series of events and carefully planned plots that followed reality, but an excuse to experiment with language itself. For instance, take the following piece of text from *Naked lunch* (Burroughs, 1959):

and

start

west

I can feel the heat closing in, feel them out there making their moves, setting up their devil doll stool pigeons, crooning over my spoon and dropper I throw away at Washington Square Station, vault a turnstile and two flights down the iron stairs, catch an uptown A train... Young, good looking, crew cut, Ivy League, advertising exec type fruit holds the door back for me. I am evidently his idea of a character. You know the type: comes on with bartenders and cab drivers,

talking about right hooks and the Dodgers, calls the counterman in Nedick's by his first name. A real asshole (...)

Furthermore, the literary layer of stories and style, which were once considered only secondary properties in storytelling, became the main concern. We can see one of the most interesting examples of these experiments in the novels of James Joyce, particularly on *Finnegans Wake* (Joyce, 1939), where grammar, syntactic and semantic rules are defied on almost every sentence:

Cry not yet! There's many a smile to Nondum, with sytty maids per man, sir, and the park's so dark by kindlelight. But look what you have in your handself! The movibles are scrawling in motions, marching, all of them ago, in pitpat and zingzang for every busy eerie whig's a bit of a torytale to tell. One's upon a thyme and two's behind their lettice leap and three's among the strubbely beds. And the chicks picked their teeths and the dombkey he begay began. You can ask your ass if he believes it (...)

Until the 19th century, traditional literary theories, such as Expressive Theory, considered a work of art as “the internal made external, resulting from a creative process operating under the impulse of feeling, and embodying the combined product of poet's perceptions, thoughts and feelings” (Abrams, 1953); in other words, a work of art in literature was considered a finished product, consequence of an individual's intentionality, which was shown to a reader whose only job was to receive the message and compare it to his own beliefs. Following this view a creative work was seen as an irreducible masterpiece that could only have emerged from the mind of a genius, which was the only person capable of creating such an original view of the world, and whose work could not be emulated.

Later on, after several decades of art experiments on the 20th century, post-structuralism argued that neither the author nor the reader creates a text, but context itself (the ‘reality’ that reader and author share) is what creates the meaning of a text (Derrida, 1967). This is possible because of the concept of intertextuality, that is, all texts are influenced by previous texts, so based only on the previously known we can find an endless discourse that can provide meaning to all future texts: everything that can be

written is already present on the common knowledge that we share as humanity. As a direct consequence of this, post-structuralists claim that an author communicates using a previously agreed symbol-code (language), which needs to meet the reader's symbol-code in order to achieve communication; meaning that even if a "completely novel work" could be done it would be useless, since there would be no reader capable of decoding it. So readers are not just passive agents, but also creative entities whose work is to construct meaning based on what is already written (Eco, 1979).

Post-structuralism gives an essential but passive role to the text itself, leaving both the writer and the reader as secondary agents. It takes away from the author the tyranny of intentionality, and as a result, the author is not fully responsible of the construction of meaning, or as Derrida would say "What text holds is no more than language" (Derrida, 1967). The meaning of a text is nothing but the intersection of the context in which it was written (including the author) and the context in which it was read, so any text holds a different meaning every time it is read. Thus, creativity could be achieved even if intentionality is not involved in the process of text generation.

There is a famous thought experiment telling that a thousand monkeys typing randomly could possibly create the Complete Works of Shakespeare, but we point out that it is the reader, and not the monkeys, who realizes about Shakespeare. Even if we claimed random production as creativity, the question would not be about the capability of the monkeys to create the works of Shakespeare, but how can we force the monkeys to produce it on a decent amount of time, since we will not live forever? In the latter case, where the amount of time counts, we would be surely talking about creative production and not just random guessing.

To finish this section, speaking again in terms of AI, we notice that the context is nothing but a huge collection of previously written text, and based on this, a way of presenting new meanings through context can be found (by a human reader) without the intervention of a human author (an algorithm could be presented as an author), since meaning is not inherent to text, but an intersection of author's context (the text corpora of an algorithm), the text (a new combination of words found on the known context) and the reader's interpretation (a human evaluator go the program's output).

3.4 A Modern Notion of Meaning

3.4.1 Language Games

One of the main assumptions of this thesis is the possibility of creation of fictional texts that are both novel and coherent. We are confident with this approach since we rely on the point of view of Ludwig Wittgenstein. This philosopher studied how is language related to meaning. (Wittgenstein, 1953). He believed that there are no such things as universal rules for the use of words and expressions in all contexts and circumstances. He said that language is not a collection of *meanings* but something that can be used to do things with it:

Think of the tools in a toolbox: there is a hammer, pliers, a saw, a screwdriver, a rule, a glue-pot, glue, nails and screws. – The functions of words are as diverse as the functions of these objects. (And in both cases there are similarities.)

Of course, what confuses us is the uniform appearance of words when we hear them in speech, or see them written or in print. For their *use* is not that obvious.

Meaning is an emergent aspect of the use of language and not an inherent attribute. This is why when we try to define words in a general fashion (e.g. dictionary definitions) the concept of *ambiguity* appears. Wittgenstein argues that ambiguities are only found in language if we detach the words and phrases from the activity or practice where they were uttered. For Wittgenstein, descriptions and demonstrations (like a scene in a play) are better ways of conveying or clarifying meaning than explanations (Blair, 2006); we should ask not for the meaning of a word but for the role that the word plays in the scene we are trying to interpret. This is what he called “Language Games”:

But how many kinds of sentences are there? Say assertion, question, and command? – There are countless kinds: countless different kinds of use of what we call “symbols”, “words”, “sentences”. And this diversity is not something fixed, given once for all; but new types of language, new language-games, as we may say, come into existence (...). The word “language-*game*” is used here to emphasize the fact that the *speaking* of language is part of an activity, or of a form of life. (Wittgenstein, 1953)

The answer to the question “What does this word signify?” should be looked at the precise instance of its use. Moreover, the meaning of a sentence is not only a matter of adding up word-individual senses, but the circumstance itself where the sentence is being used. This is precisely the phenomenon that we want to take advantage of in order to create *novel* stories: by the extraction of words’ original context, and the creation of a new one. The same series of words can mean a totally different thing if we put them in another context. From this it follows that the same syntactic structures, taken from a text corpus, correctly combined in a different way, will promote the emergence of new senses (if not meanings); and therefore, the production of entirely new stories.

Another useful metaphor from Wittgenstein is to see language as an ancient city (Wittgenstein, 1953). A place with all kind of houses: big, small, modern, antique; with an old neighborhood and the suburbs expanding more every day, and we can transit the entire city through little streets or modern avenues... According to this, language is never complete but expanding everyday, and also communication depends on how the city is traveled, with several different ways that approach the same destinations. So the routes themselves, and not just origin and destination, become ways to produce meaning.

3.4.2 Possible Worlds

Logic assigns two possible truth values to any statement: *True* or *False*; this values are translated in language into a true fact and an impossibility respectively. However, there could be two more possibilities: a fact that could have occurred but did not, and a fact that has occurred but it could also have been the case that it did not. It was from this reasoning that the notion of possible worlds emerged.

Wittgenstein stated in his earliest work that “The world is everything that is the case. The world is the totality of facts, not of things (...)” (Wittgenstein, 1921) meaning that we live in a reality that holds certain group of logical propositions, but there could be another case where one or more of those propositions are not satisfied: a possible world is then a potential reality where a certain proposition holds a different truth value (Kripke, 1980). This notion of reality is what supports the capability of creating (or just thinking of) other realities that differ from ours, having their own set of axioms, such as fiction texts.

We should also emphasize that this approach has a direct implication on more primitive facts of the world, such as the way we assign meaning to words; because if we are creating a possible world (when writing a story), then the meaning of a word is not a prescriptive meaning (such as in classic semantic theories), but it entirely depends on the other statements with which it is interacting. As we defined in Section 3.3.1, a story is a possible world created by means of language, and when a novel story is presented to a reader, a novel interpretation of each word and sentence that make that story should follow. Each possible world holds as a consequence a new set of logical propositions, making impossible to define the meaning of a text based on prescriptive word meanings; on the contrary, it forces to interpret meaning for each possible case, in each given situation, linking directly language to an activity of reality, or as Wittgenstein stated: the meaning of a word is its use in the language (Wittgenstein, 1953).

3.5 Natural Language Processing Tools

3.5.1 Syntactic Structures

Noam Chomsky introduced syntactic structures, which belong to the constituent model, in 1957. They reflect the relationships of words inside the sentence. Also, as a result of this model, Chomsky introduced the concept of Context Free Grammar (CFG). A CFG is a set of production rules that can be used to generate sentences recursively. While constituents are structures found in an already given sentence, CFG rules represent all the possible constituents (they are often infinite) that can be generated with a given grammar. So each rule of a CFG is equivalent to a constituent.

The most basic type of syntactic structure is the Part of Speech (POS) of words, which classifies words into lexical categories. Lexical categories are: Nouns (including Pronouns), Adjectives, Verbs, Prepositions, Adverbs, Determiners, Coordinators and Subordinators (Pinker, 2014). The next level of syntactic structures is the Phrase level. A phrase is a group of words; it can be of any size as long as it does not contain a subject and a verb inside the same group. There are three main types of phrases:

- **Noun Phrases (NP):** its head is always a noun and its determiners and modifiers. Examples of NPs are: “The dog”, “A tiny black dog”, and “Some extremely horrifying big dogs”.
- **Prepositional Phrases (PP):** its head is a preposition with a complement, which can be a pronoun, a NP or a subordinate clause. Examples of PPs are: “at the door”, “of the expedition whose task was to search around the planet.”
- **Verb Phrases (VP):** its head is always a verb and its complement. Verb phrases frequently have embedded PPs. Examples are “had been completely out of his mind”, “was very beautiful.”

When a group of words contains both a subject and a verb it is referred as a Clause, and finally the top level of the structure is called a Sentence, which is the root of any constituent tree.

Normally, parsers are trained with manual tagged corpora and, using this information, they generate a grammar and then try to find the most probable syntactic structure (comparing the possible structures to the already given knowledge) for each sentence. For the purpose of this work, the chosen syntactic parser was Stanford Parser, which produces constituent trees and also a special kind of dependency trees called Stanford Dependencies.

3.5.2 Stanford Parser

Every natural language can be analyzed in terms of its grammar. Linguists have developed a series of mechanisms to study languages through their structure. A natural language parser is a program that uses some of these mechanisms to analyze sentences in order to find out their structure and the role that each word plays inside the sentence.

Stanford Parser is part of the Stanford CoreNLP tools. This software uses a Probabilistic Context Free Grammar (PCFG), which is a CFG with probabilities associated to each production rule, to find the best suitable syntax structure. The parser generates the most probable constituent tree of the sentence together with the Stanford Dependency graph. This graph is a special dependency tree that can hold a more advanced set of

configurations. One of these configurations is called collapsed dependencies, which connect directly the word modified by a preposition with their dependent phrase.

3.5.3 WordNet

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations (Miller, 1995).

WordNet classifies words not only alphabetically or in terms of strings but hierarchical, considering the different senses that a single word may hold. This is why when we talk about an entry in WordNet we refer to it as a Synset instead of a word, because the relationships are made in terms of word senses; so word forms with several distinct meanings are represented in as many distinct synsets.

The most common relation between synsets in WordNet is hyperonymy (synsets that are more general than a specific one) and hyponymy (when a synset is more specific or an instance of a bigger concept). This way we can say, for example, that *dog*, *cat*, *giraffe* are hyponyms of *animal*; and *tree* holds a hyperonymy relationship with *oak*, *ash* and *pine*. Another useful relationship is meronymy where a synset is part of a whole, for example, *leg* and *backrest* are parts of a *chair*.

There are also relations on verb synsets that link them in different hierarchies such as speed (*move-job-run*), event characterization (*buy-pay*) or intensity (*like-love-idolize*). Adjective synset relations represent their membership to similar semantic uses, like antonymy (*young-old*), also relations to the nouns they are derived from (*criminal-crime*).

Relations in general are transitive within the network so there is a path between any given synsets. There are a large number of works that propose measures to determine how to measure distances and similarities inside WordNet structure. We found useful to adopt one of these metrics in order to obtain an objective semantic measure that helps to hold coherence inside our generated texts. We chose the JCN similarity measure, which we will explain next.

3.5.4 JCN Similarity Measure

The JCN similarity measure is a semantic metric based on corpus statistics and lexical taxonomy (Jiang & Conrath, 1997). It takes advantage of the hierarchical structure that already exists in WordNet and the information content (IC) that a specific word contributes in a corpus-based approach. In order to avoid polysemy problems, the measure considers not words but particular senses of words (a synset in WordNet), and it only makes comparisons among words that hold the same lexical category, meaning that a noun can only be compared with another noun, a verb with another verb, etcetera.

The JCN measure can be seen in fact a distance measure between two synsets, counting the edges linking two senses to their lowest common subsumer (lcs) or parent in the WordNet hierarchy. It is then an edge-based distance, but also adding the IC as an important decision factor.

The information content of a word or concept is derived from the co-occurrence distribution of that word in a given corpus. Following the notation from information theory the IC of a concept c can be quantified as:

$$IC(c) = \log^{-1}P(c)$$

where $P(c)$ is the probability of an instance of c in the corpus. Adding the edge-based distance with the IC information we can compute the distance between two word senses as:

$$jcn_{distance} = IC(synset1) + IC(synset2) - 2 * IC(lcs)$$

In order to transform this number into a similarity measure we can take its multiplicative inverse, so we can define the JCN Similarity Measure as:

$$jcn_{similarity} = 1/jcn_{distance}$$

3.5.5 Named Entity Recognition (NER)

Entities are specific noun phrases that represent a unique individual in the world; they can be seen as complex proper names. There are several types of entities (described in table 1). The task of NER consists on the correct isolation of entities, since they may appear in several contexts and, as we have seen, they can have any number of words. Once the boundary of each entity has been established, the NER needs to identify its type so it can be correctly referred later on any text.

Named Entity Category	Example
Organization	New York Times
Person	Doctor Bloodmoney
Location	1600 Pennsylvania Avenue
Date	Fourth of July
Time	7:30
Money	\$ 600
Percent	77%
Facility	West Australia Port
Geo-political	Victorian England

Table 1: Types of Named Entites

There are several Named Entity Recognizers, which are trained usually on news or various genre corpora; to our knowledge currently there is not an open model for NER that has been just trained on fiction texts so we decided to use the standard model from the NER that is part of de Natural Language Toolkit (NLTK) and then just adjust the results to our own purposes as we explain on Section 4.1.6.

3.5.6 Stylistics

One of the main differences between art objects and common artifacts is the presence of style. When we speak about any piece of art (e.g. a painting, a sonata, a short story) it is common to refer to the presence of *style* in the work, even it is common to talk about an

author's individual style. Style is fundamentally about the *manner* in which something is done or made, which is separate from its primary intended effect, or its *function* (Argamon & Burns, 2010).

Normally artifacts are made to accomplish a specific need, so their evaluation can be formalized and done in terms of how well they fulfill their purpose. But when we speak about art, the function of a produced artifact is far from being its most important quality, and style becomes a primary issue. Stylistics is the field of study that tries to understand the properties of manner and formalize style in terms that can be evaluated.

This does not mean that style is completely separated from function and meaning; on the contrary, manner together with function conveys meaning to observers. In fact, manner is the way an artist has to give more clues of interpretation to her intended audience; and these clues could be in the form of feelings, cultural associations, or audience emotions among others. For example, when hearing a sonata what we get depends on the artifacts (the piano and the score), but also on the manner (how hard are the keys hit, how quickly are the pedals pressed...)

Different people usually find, accidentally or on purpose, different ways to achieve the same thing, and this is why style is often related to personality. It is also common that the same person solves different problems, or creates different things using the same methods already known by her, so her manner is somehow *traceable*. On the other hand, when someone says that something “lacks of style” it is because there is not a special set of special characteristics that go beyond its main function. Finally, even if style is directly connected to personality in humans, they need to learn it some way:

Good writers are avid readers. They have absorbed a vast inventory of words, idioms, constructions, tropes, and rhetorical tricks, and with them a sensitivity to how they mesh and how they clash (...) Writers acquire their technique by spotting, savoring, and reverse-engineering examples of good prose. (Pinker, 2014)

Stylistics then, considers three levels of expression: patterns, meanings and feelings. When we try to create art or style automatically we should consider them, because creating

formal models is not enough, we need to go beyond and try to engage with the role of the reader/listener/viewer that will interpret our output, so we can be able to capture or generate text/music/images more as an experience and not just as data.

We have reviewed some of the most important concepts that are involved in storytelling, from four perspectives: philosophy, linguistics, literary theory and automatic text generation. In many of the previous works the problem of story generation has been addressed from just one of these perspectives. We have shown in chapter 2 other examples that intend to approach the problem by combining different resources from various perspectives.

Our purpose is to combine every concept that was described in this chapter to pursuit a text generation framework that overcomes the issues that previous works present. In the next chapter we will describe our proposed framework, which generates novel texts that not only keep semantic coherence, but also hold a literary style. In addition, our proposed framework will be capable of generating a different text on each iteration, even if it receives the same input; therefore, its output presents a linguistic richness that is not seen in previous works, giving the impression that an author's creativity is present in the generated text.

4. Proposed Architecture

In the previous chapters we revised a number of works related to short story generation, and we also examined the classical NLG architecture that is used to generate text in any given genre or environment. However, we can see that NLG generation, as a general purpose task, focuses more on the process of controlling structure, coherence and content; while storytelling generators deviate from the classical NLG architecture and try to overcome its limitations by developing layers that produce text with more genre-specific characteristics such as originality, entertainment, character development, story logic, etcetera.

Most of the previous work is done as a top-down approach. Following the classical NLG architecture they just add or remove capabilities to fulfill their specific goal. Besides, they strongly rely on handmade rules, knowledge bases, or schemas to overcome some of the linguistic problems that arise while generating text. What we propose is to unify both text and story generation processes through a corpus-based approach, in order to get stories that produce the sensation of literary style instead of just a carefully planned and structured text. It still imitates NLG classic architecture components, but it uses them through a bottom-up approach. Instead of generating language following a detailed document plan, we propose to start with a word as a basic unit (such as a verb, a noun or an adjective), and then let the story *emerge* based on the linguistic properties that this word holds in the knowledge base. We call these linguistic properties Literary Structures; the name comes after the fact that the knowledge base will be constructed based on a corpus made of fiction books and novels.

We propose to create a knowledge base by extracting existing relations of words and style from fiction texts. The books that conform the corpus do not matter because we do not intend to follow a specific style or author; on the contrary, we are looking for a mixture of texts that will be able to produce text with a style that is not directly referable to any previously known story. Many books are not completely present on the corpus because, for purposes of algorithm complexity, we limited to sentences equal or minor to one hundred words. We also separated sentences that are *pragmatically neutral* (sentences that do not express a question, a dialogue or quotation)¹. By using only pragmatically neutral sentences we can generalize (although there could be a few counterexamples) that the subject is the noun phrase or pronoun that immediately precedes the verb or auxiliary inside the sentence (Payne, 2011), which will help us on the step of extracting our basic building blocks (NPV, VPP, PPP or PHR). All these deletions, instead of limiting our information, help us to correctly detach even more phrases and isolate them from their original context.

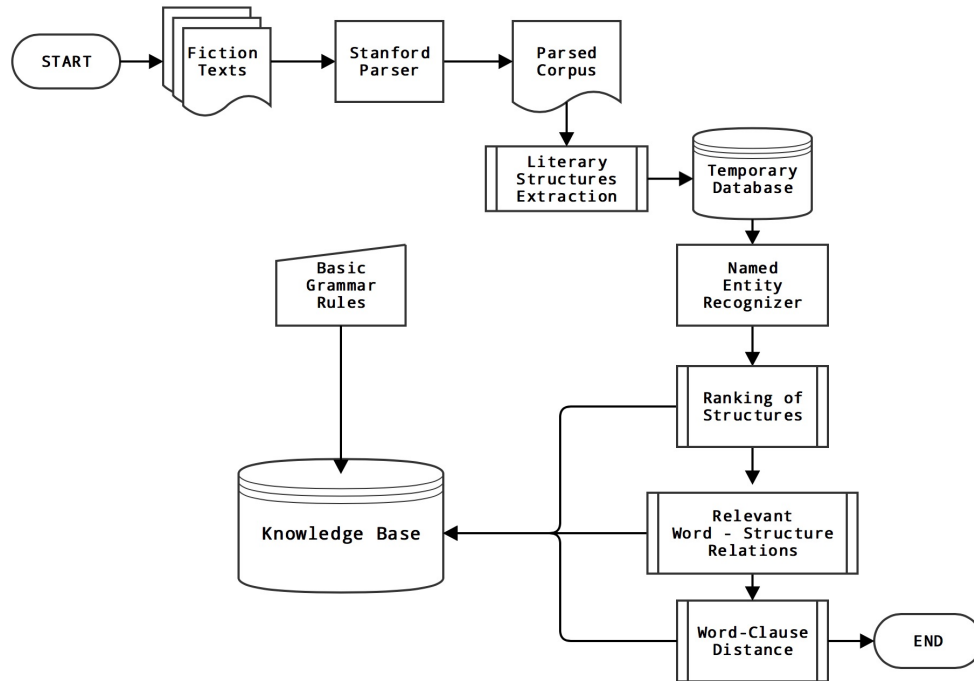


Figure 7: Flowchart of the Knowledge Base construction process

¹ We do use the *non-neutral* sentences in the Special Clause Extraction, since those are not building blocks, and they are the only sentences inserted in the generated text as they were found in the corpus, meaning that we can trust in their syntax (see Section 4.1.5).

Once the knowledge base is constructed, the first step for generating a story is providing a word to the algorithm; so it can look for the literary structures that are directly connected to that word on our knowledge base. After extracting the needed structures, the algorithm ranks and proposes a new way to relate and combine them in order to generate novel sentences that will still hold literary style, and also will produce new meanings that were not necessarily present on the original sentences. The proposed generator aims to maximize the impression of creativity, meaning that even if the content and quality of the produced sentences will directly depend on the corpus, the sequence of sentences that will conform the output will not be easily traceable in the original sources. The algorithm will combine and modify as much as possible what is found in the knowledge base without losing coherence, producing a new piece of text in each iteration: an original story.

By following this approach we are relying more on the knowledge base and less on rules and complex generation algorithms. What we accomplish by doing this is to avoid over-structured pieces of text that give the impression of not being written by humans. We also achieve to let the story meaning come out from the words, instead of carefully planning what will be said and then subordinating words to predefined contexts. We can see that by manipulating language only at the syntactic level we are able to alter its original semantic sense, exposing the bridge between syntax and semantics. Our stories can be called creative stories because on the generation step we do not code predefined rules and intentions, we only exploit linguistic capabilities of words assembling them in novel manners, resembling what a post modern human writer would do.

Besides, we dramatically reduce the complexity of language generation by putting much more effort on the knowledge base construction, converting the generation task in a simple bounded search and ranking algorithm. Because of this, once the knowledge base is built, we will be able to generate a large number of stories, all of them different from one another (even if we start always with the same word), in a very small amount of time.

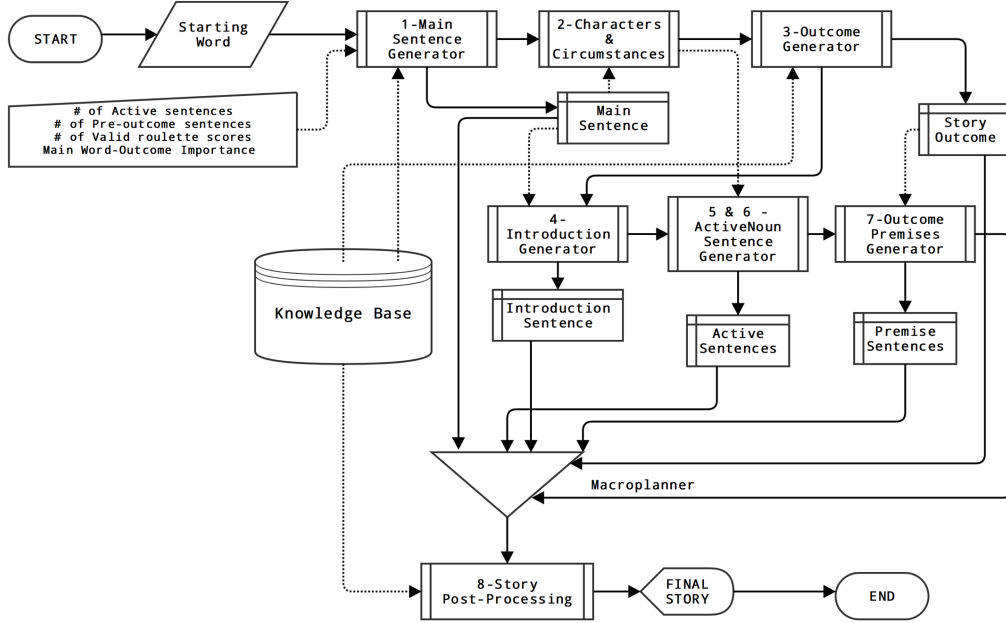


Figure 8: Flowchart of the story generation process.

Finally, we should mention that we merged the modules of Surface Realization and Microplanning, doing both tasks on the same step. Style, content, and language production will be managed as a whole, so *creativity* on the story will entirely be produced here; this way we avoid being predictable while generating sentences of the same kind or of the same length for example. We also reduced the macroplanning task to a general schema that every story will follow. This is made in order to ensure that a story with a recognizable beginning and ending will always be created. We found that this was feasible without losing the creative approach because of the diversity of sentences that can be produced on the microplanning step, which will overshadow the macro structure of the final story.

On the next sections we will first describe how is information extracted from the corpus; then the additional default information that is included in the knowledge base; next, how this knowledge is exploited to generate first coherent sentences and finally, by assembling sentences together, the generation of an output that can be called a story.

4.1 Literary Structures Extraction

In order to generate stories, first we need to build the knowledge base. We begin by processing the corpus through the Stanford Parser, so the real input that we use for constructing the knowledge base are the parsed sentences that Stanford Parser outputs.

Once we have the parsed corpus, we begin by extracting literary structures from it, which are the main components of the knowledge base and will be the basic building blocks on the generation step. We propose five types of structures: Noun Phrases with Verb (NPV), Verb Phrases with Preposition (VPP), Previous Prepositional Phrases (PPP), Simple Phrases (PHR) and Clauses (CLS). We will explain each structure separately: first we define them, then we will explain the patterns that we look for on the parsed sentences in order to extract them, and finally the purpose that each one of them has and how they contribute to the sentence construction.

4.1.1 Noun Phrases with Verb (NPV)

NPVs are the Noun Phrases of a sentence that act as agents together with the action of that agent: the main verb. Classical NPs are usually subjects or objects on the constituent tree of a sentence. As we can see on the Table 1, retrieving the subject or object as a noun phrase instead of just a noun gives us the ability to immediately know the modifiers of the noun. Also, on the dependency graph we can track the relationship that is hold between the subject noun and the main verb (see Table 2), telling us the possible actions that a noun can take or receive.

The NPV structure mixes those two properties of the parsed sentences and store the relationship subject-verb not just as noun-verb, but including on the relationship the characteristics of the subject. This way we are capable of knowing not just what actions a subject can do, but also how is the subject that is capable of doing such things.

If we take the word *woman* we can extract from the parsed corpus how women are:

Noun Phrases with “woman”
The white-haired woman
An energetic blue-eyed woman in a housedress and sneakers
His woman , a young brunette captured from a neighboring tribe
An unusually tall and attractive young woman
Only the tall black woman
The tall, full-grown tendril-less woman standing in the doorway
The barmaid, a cheerful woman about fifty years old,
Several tall, scraggly men, a woman , and two children

Table 2: Examples of possible phrases that contain the word “woman”

We developed two modes for extracting NPV structures: merging a NP with the subject-verb relationship on the Stanford graph and looking on the constituent tree for the pair NP-VP occurring on the same tree-level. On the latter case, we take only the leaves of the NP sub-tree (words) and paste the leaves of the VP sub-tree truncating it until the last verb that appears on the VP. As we can see on figure 1, we can extract two NPVs from two given NP-VP pairs:

[A master of languages]_{NP} – [would have been baffled trying to name the tongue the man spoke]_{truncated-VP}

[The man]_{NP} – [spoke]_{truncated-VP}

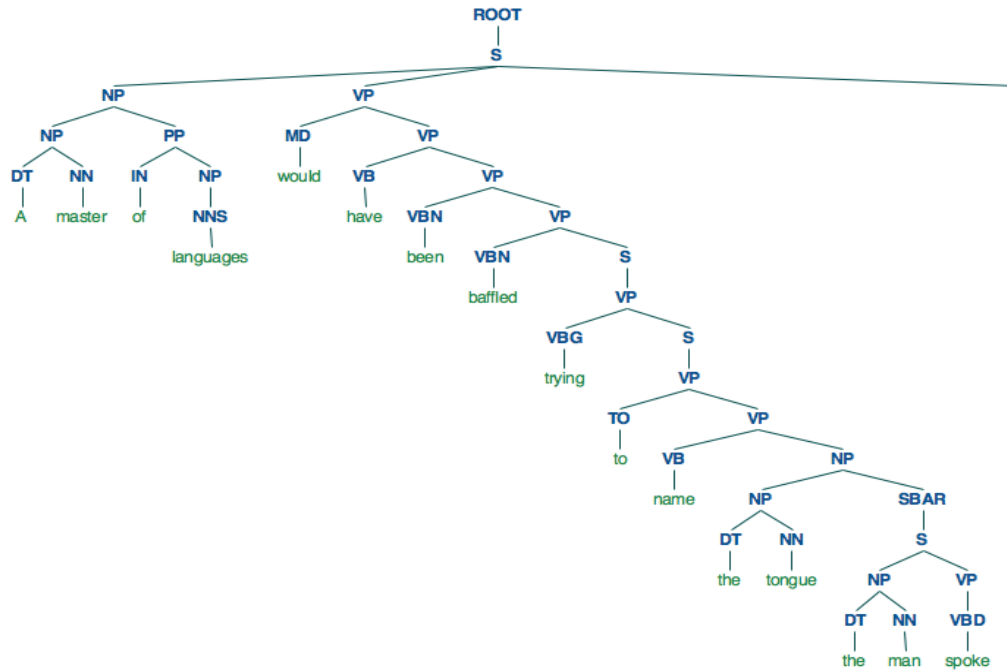


Figure 9: VPPs are obtained from the syntactic Verb Phrases found in parsed sentences.

The knowledge base then will hold a list of NPVs where all the possible actions of a specific noun, including the possible modifiers of that noun, are explained; or being more specific, we can see all the possible actions of a specific character. We can see in Table 3, for example, what can a *white-haired woman* do:

“White-haired woman” NPVs

The white-haired woman’s calmness **dashed**

The white-haired woman **looked**

The white-haired woman **made**

The white-haired woman **opened**

The white-haired woman **shivered**

The white-haired woman **waited**

The white-haired woman **stalked**

The white-haired woman **was**

Table 3: Examples of NPVs that contain the expression “white-haired woman”

4.1.2 Verb Phrases with Preposition (VPP)

Verb Phrases hold verbs with their respective predicates. They always start with a verb, and as a consequence they lack of a subject; but they clearly explain actions, objects and circumstances. Normally VPs have embedded Prepositional Phrases (which represent the circumstance of the specific action), so in order to detach more effectively the action from its context we propose the VPP structure which is the result of *subtracting* the prepositional phrase from the VPs. VPPs only keep verbs with their respective object and a preposition at the end, if it has one. We can look at the differences between VPs and VPPs on Table 3:

Verb Phrase	VPP
undertake the development and staffing of the world and its habitants.	undertake the development of
help you with whatever you have to do the last day of school	help you with whatever you have to
remained silent for a moment, and then nobody could stop him	remained silent for

Table 4: Comparison between Verb Phrases and VPPs

For the purpose of holding better sentence coherence, we do not lemmatize the verbs on the VPP, so we can look for VPPs that start with the same verb in past, present, or present progressive and we will get a different set of VPPs. For example, if we take the verb *stalk*, and search for VPPs that start with the past tense *stalked* we only get past tense instances:

“Stalked” VPPs
stalked a dozen yards away beneath
stalked about like a bristling cat ready to
stalked across the room and pushed her roughly onto
stalked away through the brightly colored sprawl of
stalked over to the bars in
stalked softly into
stalked through the open gate on

Table 5: VPPs containing the word “stalked”

4.1.3 Previous Prepositional Phrases (PPP)

Finally, the circumstances are kept the same. The Prepositional Phrases always start with prepositions and may have any predicate that matches grammatically with the preposition. We may use PPPs to attach a particular valid circumstance to a VPP if it is desired. For example, if we wish to put a preposition predicate to the previously shown phrase **“stalked a dozen yards away beneath”** the natural question would be... beneath *what*? And some of the possible answers would be:

“beneath” PPPs
beneath the rounded dome of a closely shaven skull.
beneath her own hands.
beneath the trees.
beneath this massive cylinder of a body.
beneath the thousands of padded feet.
beneath the whistles and the laughter.
beneath the surface.
beneath the makeup, and the feminine airs.

Table 6: PPPs containing the word “beneath”

4.1.4 Simple Phrases (PHR)

Literary texts hold a lot of subordinate clauses embedded in complex sentences. In order to avoid the complexity of long sentences, and also with the purpose of decontextualizing, we extract subordinates and coordinate phrases into a different literary structure named PHR, in order to be able to combine them also as separate ideas. For example take the parsed sentence “On the night he had chosen months before, Malacar Miles crossed the street number seven, passing beneath the glowglobe he had damaged during the day.”

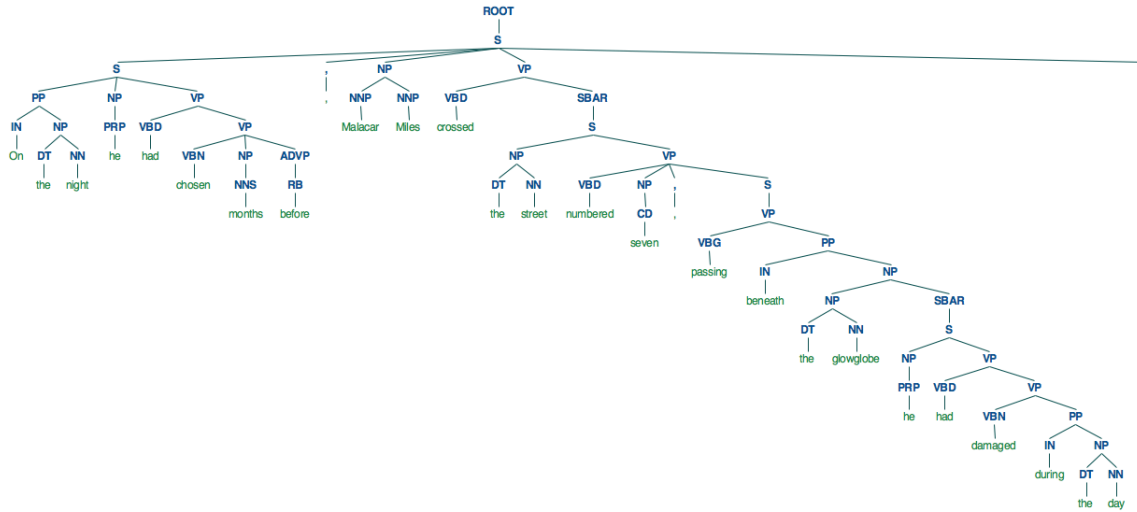


Figure 10: The constituent tree of a 25-word sentence.

We can extract four subordinates (marked as S or SBAR on the parsed tree) from one complex sentence:

Subordinates

On the night he had chosen months before.

The street numbered seven, passing beneath the glowglobe he had damaged during the day.

Passing beneath the glowglobe he had damaged during the day.

He had damaged during the day.

Table 7: Examples of syntactic subordinates found within sentences

We noticed that not every subordinate holds a complete idea or serves for our purpose of using them as building blocks for constructing new ideas. This is why we keep only the subordinates that fulfill a simple set of rules. We recognize five different types of PHRs, which will be described next.

4.1.4.1 Agent Subordinates

We call *agent subordinate* every subordinate phrase that starts with a personal pronoun. It also needs to have a verb, so we know the pronoun is participating in an action, and we just keep the phrases that end with a noun, so we can be more certain that the enunciated idea is complete. Examples of agent subordinates are:

Agent Subordinates

He saw the singularly beautiful woman.

It was, perhaps, a rather vague and sketchy plan, but at least it was something.

She assumed that she was staying in present time and in three-dimensional space.

He climbed upward yard after yard.

It had been completed to the eighteenth floor.

Table 8: Examples of Agent Subordinates

4.1.4.2 Determined Subordinates

This kind of subordinate is recognized if it starts with the defined article *the*. We just keep the phrases that have a verb inside and end with a noun, for the same reasons of agent subordinates. We found that subordinates starting with a determined article have at least a noun phrase, and if it also contains a verb, that verb tends to explain some action about the NP. Some examples of determined subordinates are:

Determined Subordinates

The man chanted hoarsely in his strange lingo.

The blade no longer looked innocent: it was a bared glittering thorn of steel.

The woman wore a tight fitting suit-like garment with red-fluorescent symbols on the breast.

The more it seemed possible that she could care for him, the more anxious the prospect made him.

The bronze masterpiece opened its mouth.

Table 9: Examples of Determined Subordinates

4.1.4.3 WH-Subordinates

This may be the more prototypical kind of subordinate. Although they don't represent complete ideas, subordinates that start with *wh-pronouns* (what, who, whom, which, whose, where, when, why, and how) are quite informative, and since it is a closed list of

pronouns we can manage them later at the generation step in order to produce complex sentences that contain one of this kind of subordinate. Examples of them are:

WH-Subordinates
When he saw the singularly beautiful woman
Which he offered to the old man
Whose task was to search for the planet
How a man's courage can turn to water
Who must be treated like a woman

Table 10: Examples of WH-subordinates

4.1.4.4 Complex Subordinates

Finally we called complex subordinates the ones that have an inner comma. We also require them to have at least one verb. They often tend to be complete sentences that appear directly on the corpus, but we decided to consider them in order to have more syntactic diversity on the generated texts. Examples of complex subordinates are:

Complex Subordinates
A moment before, this unsavory fellow had visited the washroom in the rear of the plane.
Her tent has richly fashioned, with silk-like hangings, and thick rugs and pillows.
That person found a fat man, and did a good job of masking himself in the gentleman's shadow
His thoughts focused on the dwarf, standing rigid a few steps ahead of the young man.
Our King is, as you may be aware, a sick man.

Table 11: Examples of Complex Subordinates

4.1.4.5 Descriptive Phrases

We are talking about the phrases that contain copulative verbs, which are verbs that join a subject with a subject complement (appear, be, become, look, seem). For the purpose of this thesis we will call them *descriptive phrases* because they introduce a description of

the status, or a new predicate of a previously mentioned character in the story. Descriptive phrases help to advance the story in a passive way. They are a subset of agent subordinates and determined subordinates. For example, if our main character is a dog we could include in the story one of the following:

Descriptive Phrases for “dog”
The dog was barking.
First of all, the dog became ill and was dying.
She didn't like the way the dog looked at her.
He couldn't tell whether the dog was at the other end or not.
Someone said that dog was not hurt.
The dog appeared and a stone flung out.

Table 12: Descriptive phrases containing the word “dog”.

4.1.5 Special Clauses (CLS)

Some sentences need to be entirely separated from the others because they contain specific keywords that help every story to advance its plot. Such clauses are typically present in formal texts such as essays or news to indicate an immediate cause-effect relation, but we can also take advantage of them in the narrative texts. We give the name of *special clauses* to those clauses that hold an immediate cause-effect relation, which can be found if the sentence fits exactly in one of the next pattern rules:

- The sentence starts with the word *since* or *as* and it is not part of the expression “as a matter of fact” or “as *SUBJECT* said”. Then the pattern is: **since** <CAUSE>, <EFFECT> or **as** <CAUSE>, <EFFECT>.
- The sentence starts with the word *when*. The pattern is **when** <CAUSE>, <EFFECT>.
- The sentence has the expression **because of**<CAUSE>, <EFFECT> or <EFFECT> **because** <CAUSE>.
- The sentence has one of the following structures: **If** <CAUSE> **then** <EFFECT>, <EFFECT> **if** <CAUSE>, or **If**<CAUSE>, <EFFECT>.

- The sentence contains one of the following keywords: *therefore, consequently, hence, for the reason that, as a result, as a consequence*. The pattern is **<CAUSE> keyword <EFFECT>**.

These Special Clauses (Table 9) are very important because, if we put them at the end of a story, they give the reader the impression of closure. We found that, since they hold a cause-effect relation inside the same sentence, they appear to provide a coherent conclusion to previously written sentences, especially if the clause is semantically close to the previously mentioned entities. We will see how we handle this on section 4.2.6 (Word-Clause Similarity.)

Special Clauses (CLSs)
If the woman had been nervous before → then the long wait would not have calmed her fears.
When he flattened close to a rock spine to listen → the snow sounded like sand on the snow.
As their rage increased → the pirates snarled at each other like mongrel dogs.
Because the warriors do no labor → many lazy men join the fighting group.
Since I'm supposed to be such a good loser → I'll see that nobody gets close to me and get killed.

Table 13: Examples of extracted special clauses.

4.1.6 Substitution of Entities on Literary Structures

It is important to mention that we have a middle step between the extraction of the structures and their storing. We have a temporary database where the extracted phrases are kept with all the Entities from the original text: proper names, locations, private organizations, and etcetera. Once we have processed the whole corpus, we put the structures through the NER from the NLTK library to identify where every entity is. We used the standard model that comes with the recognizer; for this reason, the level of correct matching was not successful enough. In fact, after utilizing the standard

recognizer, we corrected most of its mistakes by only keeping the most frequent tag associated to a recognized entity. This was a good approach, since entities that are mentioned a lot in a text, for example the proper name of the main character of a novel, appear a few dozen of times, so even if the accuracy of tagging is mediocre, it is expected that the most frequent tag attached to it will be reliable.

4.2 Knowledge Base Implementation

Now that we described the kind of structures that were extracted from the corpus, we need to explain how we organized and create relations between them in order to optimize the generation step. But before doing this, we also describe a series of *default* vocabulary and linguistic properties that our generator needs to *know* in order to produce grammatically correct full sentences. It is true that by extracting structures we preserve most of the syntax correctness, but by giving some basic linguistic knowledge we will be able to manipulate the produced sentences even better, improving the coherence of the stories.

We call it default knowledge because it implies some abilities that we as humans take for granted when producing language but need to be taught to an algorithm. There is also another kind of default knowledge about writing stories, for example common story introductions or a common way to introduce a character in the scene. Since in most of the cases this default knowledge can be classified in closed sets of characteristics, we segmented it in different lists so it will be easy to identify, at the time of producing text, if we are using a word or an expression that fits in one of this categories, and then we can treat the case following the conventions that being a member of a set implies.

4.2.1 Transitive and Auxiliary Verbs

Transitive verbs are those which need an object or predicate after enunciating them. We have a list of the 380 most common transitive verbs. This way, if we find that we produced a sentence that finishes on a transitive verb, we know that we need to find a suitable predicate that complements the verb in order to produce a sentence that holds a complete idea. Also, it is important to mark which verbs are functioning as auxiliary or modal, this

is useful to keep number and gender agreement when conjugating sentences at the generation step.

4.2.2 English Prepositions

Prepositions in English function as syntactic connectors of VPs and NPs. They go beyond syntax and express several kinds of semantic roles (Payne, 2011), so their correct use is of great importance. For this reason we included a list of 70 prepositions, to identify if some sentence is leaving an unconnected preposition.

4.2.3 English Nouns and Pronouns

Generally speaking, nouns in English do not hold gender. However, we can find several cases that refer only to feminine or masculine entities, this is true when talking about nouns that name feminine roles (e.g. *actor/actress*, *boy/girl*), feminine animals (a masculine member of cattle is *bull*, and a feminine member is *cow*). Another characteristic that is important to take into account when using nouns is the recognition of singular and plural forms since not all plurals are regular (especially when words come from other languages like Latin e.g. *hypothesis/hypotheses*, but also more common words such as *child/children*).

All these become important when producing texts with more than one sentence, because of the phenomenon of Referring Expressions, which is usually managed by the microplanner in Classic NLG, but as we are merging microplanning with language production we need to attack the problem directly. We decided to have lists of the most common nouns that are exclusively feminine or masculine, also a list of irregular plurals, and finally to attack the problem more efficiently we rely on a NLTK WordNet module called Morphy to properly identify gender and number in nouns.

Once we properly identify a noun it is easier to know which quantifier to use if one is needed, or which pronoun should we use to refer to a previously mentioned entity, this is done to avoid excessive utterance of the main characters or places.

Pronouns are used in English as a substitute of previously known NPs, so it is important to know the gender and number of the NP to use the correct pronoun as its substitute, this is why we keep default lists of pronouns so the correct substitution can be made at the generation step. Also we keep a list of relative pronouns, also known as wh-subordinators, which often help to identify relative clauses. This is important not just to properly identify some of the structures that will be extracted from the corpus, but to properly integrate relative clauses at the generation step.

4.2.4 Proper Names

Proper Names are very common in storytelling. In fact, many famous stories can be instantly identified by the names of their characters. This is why we decided to erase the original proper names from our building blocks for the stories that we will generate, in order to avoid using the names to link fragments to previously known stories. So we also included in the knowledge base a list of the 100 most popular names for girls, and 100 most popular names for boys in English in 2014². So whenever a proper name is needed in a story, since these names are very popular, they will not influence directly the meaning of the produced story, nor easily link a situation to a previously known one.

4.3 Knowledge Base Navigation

Now that we have described the structures that are extracted from the parsed corpus, we should explain the way in which they are connected. The knowledge base consists on a main *<key,value>* word database, where all the unique words are the keys and the value is a concatenation of the IDs of all the structures (phrases) that contain that word. Also there is a separate *<key,value>* database for each type of structure (NPV, VPP, PPP, PHR, and CLS) where the key is a unique ID and the value is the phrase that compose the literary structure. Additionally, word and structure frequencies are stored to contribute to the phrase-ranking step.

In order to ease the visualization of the knowledge base, this series of databases can be seen as a graph with phrase nodes and word nodes, where an edge between a phrase node

² Taken from the website <http://www.babycenter.com/top-baby-names-2014>

and a word node exist if the word is mentioned in the phrase. For the purpose of reducing complexity we only connected the nouns, adjectives, prepositions and verbs inside phrases with their respective word nodes. If we see the knowledge base as a graph we can say that the construction of a phrase is a simple navigation on the graph, where an initial word is asked as the main key and a sub graph is obtained with the main word as the only word node together with all the phrase nodes that have an edge to that word (see Figure 3.)

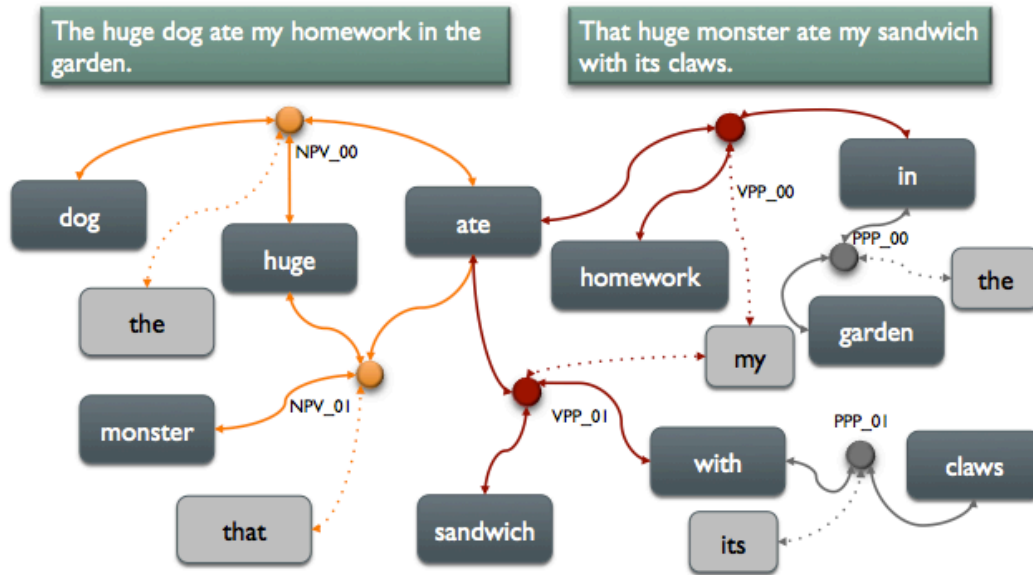


Figure 11: A graph visualization of the knowledge base showing NPV, VPP and PPP connections

4.3.1 Sentence Construction

Once we have the corresponding sub graph of the main word, we have all the building blocks that are necessary to construct novel sentences. We propose to construct novel sentences based on a simple pattern as shown on the next figure:

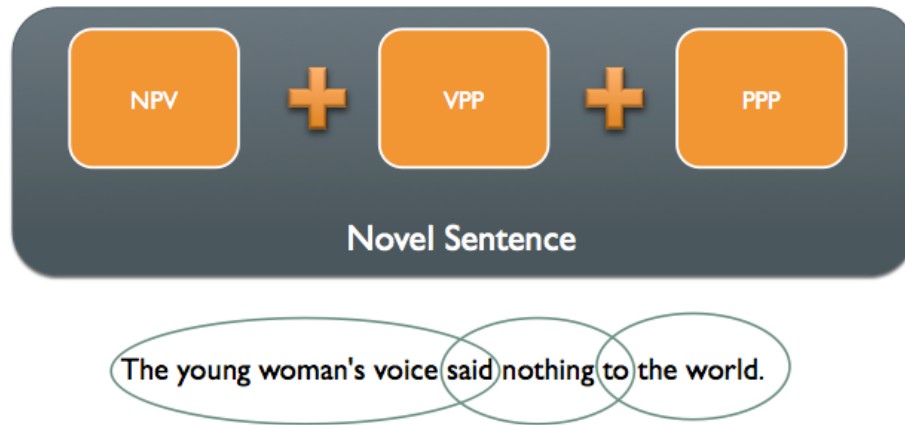


Figure 12: Novel sentences are concatenations of intersected literary structures

We can see that the proposed literary structures can be easily concatenated because of their defined properties. Given a word we can have a large number of combinations among the connected phrases. We can start by choosing an NPV, and then we can look for all the VPPs that start with the same verb as the NPV ending verb, and the same can be done with looking for the PPPs that start with the same preposition as the ending preposition of the chosen VPP. The next section will explain which structures are preferred and then how are they chosen.

The key idea is that, by following this simple approach, we can easily expand phrasal possibilities, starting from a main noun, into several different characters (Figure 5), then by looking at VPP intersections we can expand even more from the main verb into several predicates (Figure 6), and finally if we want a complex predicate we expand the preposition into many prepositional complements (Figure 7). We have also the option of just producing sentences that are formed by NPV+VPP, omitting the PPP expansion and truncating the VPP until the verb; in case that the verb is transitive, we just look for a suitable predicate that completes the idea.

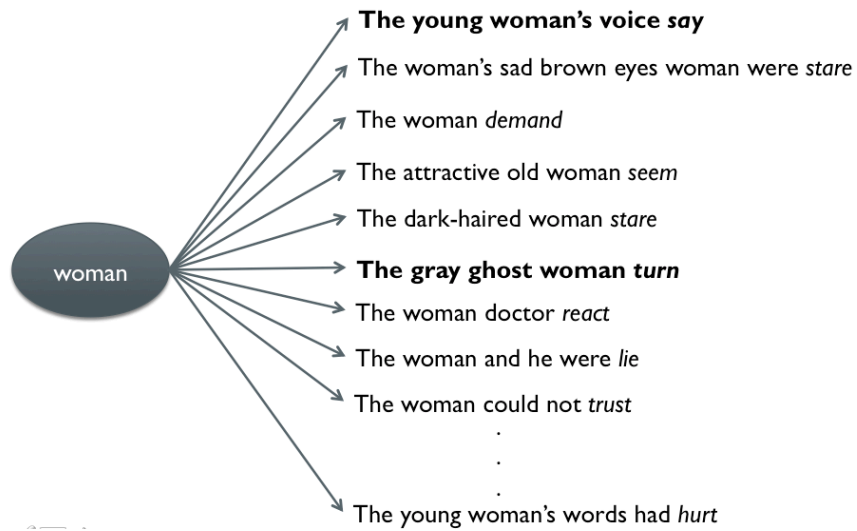


Figure 13: Examples of NPVs connected to the node word *woman*

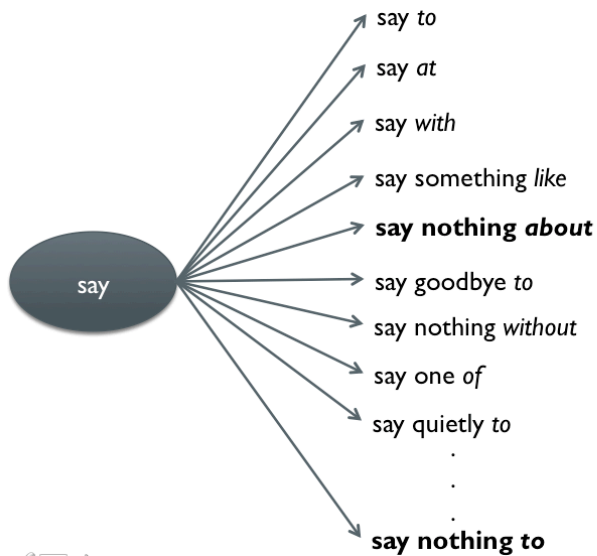


Figure 14: Examples of VPPs connected to the node word *say*

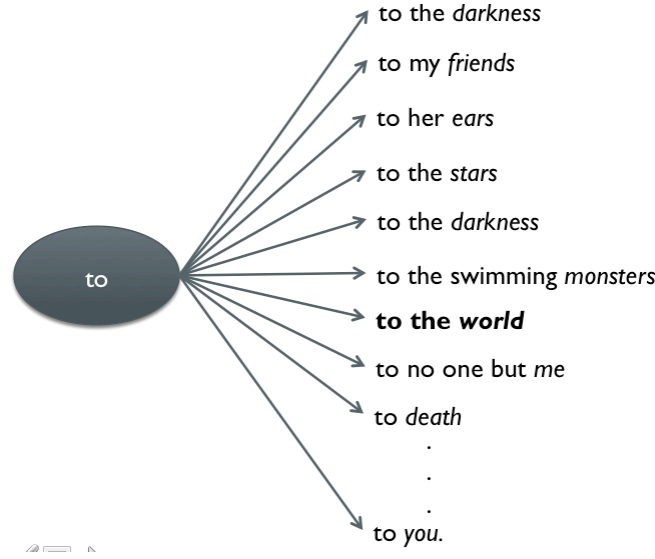


Figure 15: Examples of PPPs connected to the node word *to*

4.3.2 Word-Clause Similarity

A separate database is kept for storing semantic similarity between words and special clauses (CLS). For now we will just explain that the values are pre-calculated, this is done to save time at the generation step, and we will fully explain the purpose of storing these values on Section 4.5.3. This database contains the most frequent 200 nouns, adjectives and verbs (a total of 600 words) as the keys, and its values are the concatenation the semantic similarity score, which we call *sensim*, between that word and every CLS that was found; this is obtained with the help of the JCN Similarity Measure. With this score, given a word, we can know which are the semantically closest CLSs to that word.

We build a database of similarity measures by comparing each of the 600 words to every stored CLS on our knowledge base. As we mentioned on section 3.5.4 when we explained the JCN measure, comparisons can only be made between two words with the same lexical category, so to achieve a comparison between a word and a phrase, we sum the *sensim* between the main word W and each of the n words inside the clause whose lexical category matches that of the main word:

$$CLS_sim(W, CLS) = \sum_{k=0}^n sensim(W, w_k)$$

We use the operation *sensim* because JCN measure works with *word senses* instead of words, and since words have been detached from their original context, it is quite difficult to trace their sense back, so we propose to analyze the similarity among every sense of both words and keep the biggest score found (that means we keep the farthest semantic similarity that two words can have) this operation can be described as:

$$sensim(w_1, w_2) = \operatorname{argmax} \left\{ \sum_{i=1}^x \sum_{j=1}^y jcn_{similarity}(s_i, s_j) \right\}$$

Where x is the number of senses from w_1 , and y is the number of senses from w_2 .

4.4 Phrase Ranking Algorithm

We have said that a huge number of sentences can be built by freely combining all possible NPV+VPP+PPP structures. However, in order to get better syntactic coherence in our generated sentences, we defined a set of simple ranking rules for each of those three structures. This rules were based on semantic and style manuals, but the scoring and combination of rules, as well as the manner in which they are applied are a direct contribution from us.

Each list of possible structures will be ordered according to a score that our ranking algorithm assigns to them. Every existent structure starts with a score of 1, and the points will be added depending on the rules that we describe next. Finally, to avoid losing the creative feature, we run the roulette wheel method in order to choose them; so the higher ranked structures will have more probability to be chosen than the poor ones, but each one of them has a probability of being chosen anyway. We will explain separately the aspects that our algorithm considers to rank each structure, and also how the roulette method works in order to choose the phrases.

4.4.1 NPV Ranking

We are using NPV structures as the subjects for our generated sentences. For this reason, we prefer determined noun phrases because they tend to appear as subjects on the corpus. Also undetermined noun phrases (which start with *a/an*) are preferred, because there is no syntactic difference if we substitute the undetermined article for a determined one. We also penalize phrases that present evidence of being subordinates, because placing this as subjects generate semantic incoherence. Also the algorithm gives preference to NPVs that contain more adjectives, this is because of two reasons: more adjectives give us more opportunity to expand further sentences, and also a longer NPV holds a style closer to literature since it gives the impression to be more expressive. Finally, we add to the score the frequency multiplied by two that each structure had in the corpus. We list the complete set of rules here:

Feature	Score
NPV starts with “The”	+500
NPV starts with main word	+300
NPV starts with “A/an”	+300
NPV has one or more commas	+50
NPV has a subordinator	-100
NPV has the particle “to”	-300
NPV verbosity	+ (Number_of_adjs * 10)
NPV frequency	+(NPV_freq * 2)

Table 14: Ranking algorithm rules for NPV literary structures.

Here is an example of how an NPV score is assigned. Take the NPV structure “The old woman, small and grey as a mouse, never said”. The score starts in 1, then we add 500 points because the structure starts with “The”; next, we add 50 points because it contains a comma; the verbosity of the sentence adds 30 points (because the NPV has three adjectives: *old*, *small* and *grey*); finally, suppose it has a frequency of 2 in the corpus. The final score then is computed as: $1+500+50+30+4 = 585$ points.

4.4.2 VPP Ranking

VPPs are ranked a little bit different than other categories, because they strongly rely on the already chosen verb of the NPV, and we know that they should always end with a preposition. For these reasons, we penalize VPPs that do not finish with a preposition, we subtract points to nouns on VPPs because they mean more characters to handle and augments story complexity, and we also penalize commas, because they often mean that the VPP contains two or more different statements, and that may result on semantic incoherence. Finally, we add to the score the phrase frequency multiplied by two. The features considered for ranking VPPs are:

Feature	Score
VPP has a comma, semicolon or colon	-100
VPP ends with identified preposition	+300
VPP ends with determiner, noun, adjective or verb	-200
VPP verbosity	- (Number_of_nouns * 10)
VPP frequency	+(VPP_freq * 2)

Table 15: Ranking algorithm rules for VPP literary structures.

4.4.3 PPP Ranking

PPP structures function as compound predicates for our phrases. The ranking of PPP is merely syntactic: we prefer them to finish with nouns and to have more adjectives, and we penalize the ones that finish with an adjective or a verb, because it frequently means that the idea communicated will be incomplete. And, as we the last two structures, we add the phrase frequency multiplied by two. On table 17 we show the features considered for this category:

Feature	Score
PPP ends with a noun	+100
PPP ends with an adjective	-200
PPP ends with a verb	-200
PPP ends with a determiner	-200
PPP ends with the particle “not”	-200
PPP has the particle “to”	-200
PPP has a comma, colon or semicolon	-50
PPP verbosity	+ (Number_of_adjs * 10)
PPP frequency	+(PPP_freq * 2)

Table 16: Ranking algorithm rules for PPP literary structures.

4.4.4 PHR Ranking

We recall that PHRs have already been sub-categorized on the knowledge base: descriptive phrases, agent subordinates, determined subordinates, wh-subordinates, and complex subordinates. So the algorithm directly looks for a specific sub-category, which has already been filtered; however, there are few more general features that are scored to obtain more expressive phrases. These features are:

Feature	Score
PHR ends with a noun	+200
PHR has desired word as subject	+200
Determined PHR starts with a determiner	+100
WH-PHR starts with a subordinator	+100
PHR ends with the particle “not”	-300
PHR has the particle “to”	-300
PHR verbosity	+ (Number_of_adjs * 10)
PHR frequency	+(PHR_freq * 2)

Table 17: Ranking algorithm rules for PHR literary structures.

4.4.5 Roulette Wheel Selection

The roulette wheel selection is a common technique used in the field of genetic algorithms as a simple selection method where even the least fitted individuals can get a proportional chance of being chosen. It can be defined as the relationship between an individual's fitness and the sum of the fitness values of all the remaining individuals as expressed on the following equation (Alba & Dorronsoro, 2008):

$$p_i = \frac{fitness(i)}{\sum_{j \in Neighborhood} fitness(j)}$$

By doing this we define a probability distribution, where the most fitted individuals have the larger probability and it decreases proportional to the individuals' fitness. As our intentions are of exploiting the creative properties of our corpus, we saw an opportunity to benefit from this method, since we need a way to choose the best possible options for constructing our story sentences, but at the same time we want to cautiously *explore* the possibilities of putting together some pieces of text that are not considered optimal. We developed a ranking algorithm that works as a fitness function and, by using with the roulette method, we can manage to create a different story on every iteration of the algorithm, even if we input the same word all the time. To avoid rigid stories, we do not always seek for the same syntactic matches from a given word; conversely, we seek for any syntactic sentence that is able to hold new semantic properties.

4.5 Story Generation

We have established how we extract, classify and rank all the necessary building blocks for the generation step. We emphasize the importance of having a well-structured knowledge base for optimizing the text generation layer. On this section, we will mainly explain how the algorithm uses its knowledge base in order to construct a coherent story.

First, we will explain what we use as story macroplanning, which is nothing but a general template with fixed slots that will be filled, with the help of a few directives, at the end of the process. This template is filled step by step, and every step will be explained in detail

together with the information that is tracked, for purposes of coherence, at the moment of generating a story. Before putting everything together, we make some grammar corrections, which we call *Story Post Processing*. At last, we ensemble everything together, and show the final output of the algorithm.

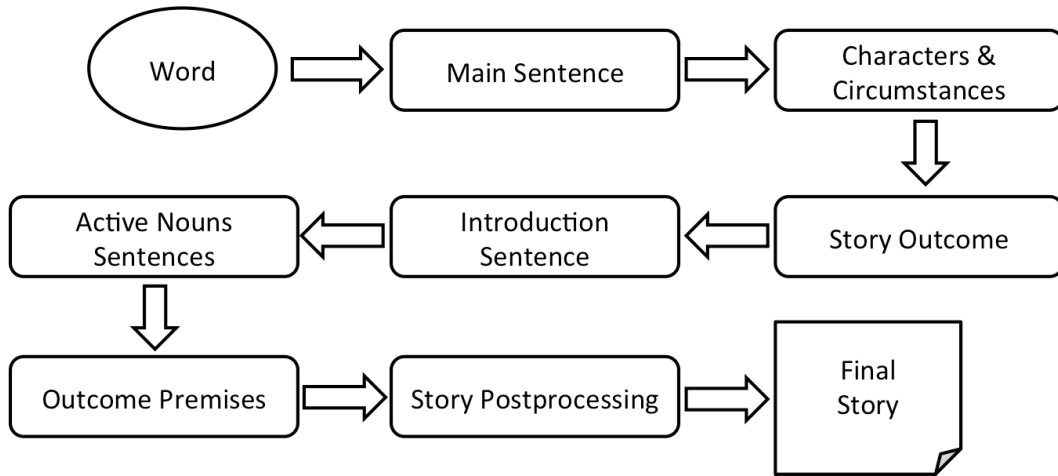


Figure 16: Steps that the algorithm follows to construct a story

We arranged the Story Generation steps in the same order as the algorithm produces them; they do not necessarily match the order in which they will appear on the final story, but they do reflect the actual logic that is followed to link ideas (see figure 12). On every subsection a general explanation of the corresponding step will be given, followed by an example taken from the working algorithm to illustrate how an actual story is constructed. Hence, at the end of this section we will see a complete artificial story, with the whole process of how it was constructed.

4.5.1 Main Sentence

The only input needed to start a story is a single word; it can be a noun, a verb, or an adjective, and it will be referred in the future as *main word*. This given word is the only thing the algorithm needs in order to find out what it knows about it, generate an original sentence, and build a novel story. The main sentence is not only the first step, but also the core of every story. Based on the input word the novel sentence is constructed (NPV+VPP+PPP as explained on section 4.3.1). The main word will be considered as the

main character of the story, and the other nouns that appear after the verb will be considered secondary characters.

If we take the adjective *strange* as our input, the algorithm will look for all connected phrases to that word. Having this subset of phrases, a filter is applied to obtain a list of NPVs, which will be then ranked. Here is a list of the top-10 ranked NPVs for *strange*, together with their score in brackets:

The **strange**, speechless uneasiness that was perceptible under his mute indifference almost terrified (583)

The **strange**, speechless uneasiness that was perceptible under his mute indifference terrified (583)

the fine features of the one whose **strange**, seductive, and achingly beautiful sounds had drawn (583)

the sound of the horn: high and terrible, yet of a **strange**, deadly beauty travels (583)

The air was damp and fetid, smelling of swamp and decay, **strange** for where they were (583)

The banker sounded distant, somehow **strange**, as if wary or bewildered (583)

The effect was **strange**: the sudden darkness, the abrupt illumination - while the <<PERSON>> whipped (583)

The furious color faded from his cheeks, and his **strange**, hot vitality go (583)

The furious color faded from his cheeks, and his **strange**, hot vitality seemed (583)

The mere fact that they were drifting helplessly on a **strange** ocean, surrounded by unknown monsters, seemed (583)

We should remember that these are just the top-ranked NPVs, but the final selection will be made upon a much bigger list by following the roulette method. This time, the next NPV was chosen: *the white sky met the white ground in a strange world where the grubby black bus station floated*; now the algorithm will consider all VPP's that start with the verb

floated, and will properly rank them. Again, for the sake of the example, we only list the top-10 ranked VPPs:

floated about (303)
floated above (303)
floated across (303)
floated across to (303)
floated after her into (303)
floated alone in (303)
floated and rearranged themselves on (303)
floated around upside down or bounced about on (303)
floated as though created mostly of (303)
floated at lumbered northward at (303)

Again, the roulette method is used to pick a VPP from the given distribution set. In the current example the chosen VPP was *floated up into*. Now we have the sentence *The white sky met the white ground in a strange world where the grubby black bus station floated up into*; and a correct complement for the preposition *into* needs to be found. Here we show the top-10 ranked PPPs that start with *into*:

into a well-tended patch of blue and yellow flowering groundcover surrounding islands of taller lace-lady ferns and monarch roses (153)
into a domed many-leveled building covered with a glistening and iridescent translucent skin (153)
into a large room full of long tables cluttered with electronic gear (153)
into a warm room with an enormous high down bed against the far wall (153)
into hydrokinetically-maintained organic tissue of what had once been-or convincingly appeared to be-a human being (153)
into the building a small brightly lighted foyer of polished white plaz and only banks of comeys (153)

into a big old-fashioned room with built-in china closets and a lot of shabby furniture (153)

into the building a small brightly lighted foyer of polished white plaz and only banks of comeys (153)

into a big old-fashioned room with built-in china closets and a lot of shabby furniture (143)

into a central interior area dominated by a single large statue in the area's center (143)

Same as with the last two structures, the roulette method is used with the distribution of PPPs to choose one. This time the algorithm chose *into a well-tended patch of blue and yellow flowering groundcover surrounding islands of taller lace-lady ferns and monarch roses* we now have a complete main idea and can go to the next step. The main idea in this iteration is *The white sky met the white ground in a strange world where the grubby black bus station floated up into a well-tended patch of blue and yellow flowering groundcover surrounding islands of taller lace-lady ferns and monarch roses*. Note that, since our main word is the adjective *strange*, we need to design a main character, which the noun that is modified by that adjective (the most immediate noun at the right of the word *strange*). As a result, the main character of this story is *world*.

4.5.2 Handling Characters and Circumstances

We have stressed in several sections the importance of adding the effect of creativity in the algorithm. This means that many decisions taken at the generation level are taken stochastically. For this reason, it is impossible to anticipate the words that will appear on the next step of the algorithm, but it is possible to keep track of all the previously mentioned words. From this it follows that future sentences can profit from what has already been said and also avoid to semantically deviate too much from it.

On every step of text generation, we add to a small dictionary the different participants of the current plot (nouns) with their specific characteristics (adjectives), and the actions they have taken (verbs). By doing so, we avoid to present the same character with contradictory adjectives (e.g. if we already said that a woman was young, we can not refer to her as an old woman on latter sentences). Also, if a noun has verbs attached to it, we know that it is

an *active* noun in the story and can be considered an agent who is able to perform more actions on the future. The dictionary will handle every entity as it appears on the story, and will add it in a structure such as follows:

```
StoryEntities = {entity_1: [[modifier_1, modifier_2, ..., modifier_p],
[action_1, action_2, ..., action_q]],
...
entity_n: [[modifier_1, modifier_2, ..., modifier_x],
[action_1, action_2, ..., action_y]]}
```

The verb attachment to nouns is done by looking in the linear sentence (the string of words) for the first noun that appears at the left of a verb, if no noun is found, the verb is omitted; if a noun is found, the verb is attached to the action list of that noun. The adjective attachment is done similarly; this time by looking for all the adjectives that appear at the left of a given noun (until another noun is found, or the beginning of the sentence is reached), these adjectives are directly attached to the given noun.

For example, take the previously generated sentence from the algorithm. The dictionary isolates the nouns of the sentence, in this case: *sky*, *station*, *groundcover*, *ground*, *ferns*, *bus*, *patch*, *world* and *islands*. Then, the algorithm looks for the modifiers of each noun, followed by the actions in which the nouns are involved. Our dictionary has nine entrances with their respective modifiers and actions as follows:

```
StoryEntities = { sky: [[white], [met]],
station: [[-], [floated]],
groundcover: [[blue, yellow, flowering], [sorrounding]],
ground: [[white], [-]],
islands: [[-], [-]],
ferns: [[taller, lace-lady], [-]],
bus: [[grubby, black], [-]],
patch: [[well-tended], [-]],
world: [[strange], [-]]}
```

From this, we can determine which of the nine nouns that are present on the main idea are actually capable of performing actions, or *active*. According to the information listed above, the entries *sky*, *station* and *groundcover* have a list of verbs attached to them; therefore, *sky*, *station*, and *groundcover* are considered as the secondary agents of the main idea.

4.5.3 Story Outcome

Before elaborating more about our known characters and circumstances, we need to know at what outcome are we aiming, in order to control a little more the semantics of our sentences. Thus, the second step is to look for a suitable story ending. Once the main idea is created, and based on the vocabulary that is present in it, the algorithm looks for the twenty closest clauses to the main idea; we obtain this by summing the similarity scores of the main word to each CLS and also the similarity-scores of each word that forms the main idea. A parameter *alpha* is added to the main word, to give more importance and priority to the clauses that are closer to it. We found that assigning $\alpha = 2/3$ gives good results. The complete formula that computes the similarity between each CLS (or possible outcome) and the current main sentence is defined as follows:

$$Outcome_sim_k = \alpha * CLS_sim(W_{subj}, CLS_k) + \frac{(1 - \alpha)}{n} \sum_{i=1}^n CLS_sim(w_i, CLS_k)$$

Once we have calculated the similarity of our main sentence to all the possible outcomes, we obtain the semantically ranked list of CLSs, and we choose the top-20 ranked outcomes (the twenty closest outcomes to our main sentence.) Finally, the roulette method is applied to choose among the top-20 list what will become our story outcome. Once it is determined, it will not change in the current iteration, so the sentences that will be produced in the story will take into account not only the main idea as the starting point, but also the story outcome as the goal.

We will continue explaining our example at this step. We recall that our main idea is *The white sky met the white ground in a strange world where the grubby black bus station*

floated up into a well-tended patch of blue and yellow flowering groundcover surrounding islands of taller lace-lady ferns and monarch roses. Our main character in this case is *world*, and the rest of the relevant words (adjectives, nouns and verbs) that contribute to semantic similarity are: *white, sky, met, white, ground, strange, grubby, black, bus, station, floated, well-tended, patch, blue, yellow, flowering, groundcover, surrounding, islands, taller, lace-lady, ferns monarch, and roses.* Note that if there are repeated words in the sentence they contribute to the index for each instance of it. We will look for the top-20 closest clauses to all of these words combined, following the provided formula. We show here only the three closest clauses to our current example:

because it was an Earth alone → He had been mightily interested in Earth's ancient past in his younger days, it was an attractive study to many Earthmen-an Earth supreme.

Since both the earth and man are sentient beings, their emanations coincide → or rather, the earth has all the emanations present in man and all the emanations that are present in all sentient beings, organic and inorganic for that matter.

It might have a real basis in fact, too, but the real reason is that we feel that a world with tigers and orangutans and rainforests and even small unobtrusive snails in it is a more healthy and interesting world → for humans (and, of course, the tigers and orangutans and snails) and that a world without them would be dangerous territory.

In this example the chosen outcome is *Since both the earth and man are sentient beings, their emanations coincide or rather, the earth has all the emanations present in man and all the emanations that are present in all sentient beings, organic and inorganic for that matter.*

4.5.4 Introduction Sentence

As we have said before, stories nowadays do not necessarily hold a classical structure of introduction, conflict, and resolution; even though, we believe that it is more intuitive for a reader to identify a text as a story if it begins in a classical manner. This is why, we chose to have a few common story introductions, in order to let the reader immediately realize that what they are about to read is intended to be a story. We are aware that there are

several techniques to start stories (in fact, there are infinite ways to start a story), but for the purpose of this work we chose five different *ways* to start, which are the templates listed in Table 14.

Story Introduction Templates

Once upon a time there was <NP>.

Let me tell you a story about <NP>.

There was a time when <NP> existed.

There's an old story that tells about <NP>.

There was once a <NP>.

Table 18: The five possible starting sentences of the stories.

A template is chosen randomly on each algorithm's iteration, so even if the same idea was produced at a different attempt, it is likely that the story starts in a different way. It can be seen that the only thing that the template needs is a NP, which is taken from the current *main idea* to fill the template and produce the actual introduction sentence.

In our example, the second case was randomly chosen, and our introduction sentence will be: *There was a time when a strange world existed.*

4.5.5 Characters Description

We explained descriptive phrases on section 4.1.4.5. The algorithm will use them in order to deepen more into the characters of the story. At this step we already have the introduction and the main idea of the story together with the story outcome; what is expected to go next in a story is a little more information about the characters, so the algorithm looks for descriptions of the main word and the other words that appear to be active on the story. We consider a word as *active* if it is a noun and it has at least one verb (action) associated to it in our dictionary of characters and circumstances. Since the number of active words depends on the main idea content, the number of descriptive phrases will vary from story to story.

In addition to the active nouns, the algorithm also looks for the noun that holds more connections (a connection is a link to a phrase that mentions the given word) in the

knowledge base, in order to write another sentence about it (since it has many connections there should be plenty of things to say about it.) It is designed as the Secondary Character of the story.

The rest of active nouns may or may not be mentioned in more descriptive phrases. A random number is generated between zero and the number of active nouns, to determine how many more descriptive phrases will be included. This is done because there are many cases where the list of active nouns is big, and in order to avoid having a bunch of phrases mentioning them and deviating too much from the main idea the solution is not to mention all of them.

On the whole: the third sentence in every story will be a descriptive phrase mentioning the *most connected* word in the main sentence. Next, a descriptive phrase about the main word will be included as the fourth sentence. And last, descriptive sentences mentioning at most each active word (or maybe zero, depending on the random number of the current iteration) will be included.

In the case of our example, the most connected noun from the main sentence is *patch*. A selection process among descriptive phrases is done, exactly as explained earlier in the cases on NPVs, VPPs and PPPs. In our example the chosen sentence that describes the word patch is:

The shadow within the white **patch** was brighter than any star outside it.

The next sentence in the story is a descriptive phrase about the main word; the algorithm chose the next one:

The underwater **world** was so very strange, like another planet.

For the active nouns, currently there is a list of three words: *sky*, *station* and *groundcover*. A random number between 0 and 3 is generated to decide how many active words will be included. In our example the resulting number was 2. Thereby, the first two nouns from the list will be described in the story. In this case the chosen sentences were:

The **station** was a bright dot in the distance, bracketing the invisible five-space jump point.

The **sky** was dawn, and the sky was changing from black to murky gray.

4.5.6 Outcome Premises

To advance even more the plot, after we described more about the characters, we have to look for a connection between the beginning of the story and the outcome (which we happen to know already!) Based on a similar technique as the character description, a separate entity dictionary is created for the outcome alone, and the algorithm now looks for the nouns that appear on the selected outcome, and tries to say more about them. We do this by assembling together different kinds of PHRs to create more syntactically complex ideas. These resulted in a good bridge between the main character and the possibly different main nouns that appear on the CLS that was chosen as the outcome of the story, giving fluency from the beginning to the story ending.

The same as with character descriptions, we decided to avoid saying too much about nouns that deviate from the main plot. Consequently, again the algorithm produces a random number between zero and the number of nouns in the list, to choose which nouns will be described before the outcome. In contrast with the character description, this time the nouns described are not the first on the list, but they are also chosen randomly. In other words, first a random number (lets call it r) is produced to decide how many nouns will appear, and then from the list of nouns r indices between 0 and the number of nouns are chosen to extract those nouns from the list. This will be explained more clearly in our example.

First, we recall our example: *Since both the earth and man are sentient beings, their emanations coincide or rather, the earth has all the emanations present in man and all the emanations that are present in all sentient beings, organic and inorganic for that matter.* We can see four new entities here: *earth*, *man*, *emanations* and *beings*. The random number at this iteration was 3; therefore three nouns will be explained. The three chosen nouns from the list were: *earth*, *man* and *beings*. So we will insert a new statement about them. The chosen statements on this example were:

pawed the **earth**, lowered his head and drove with incredible force at the older one

was thirty odd, almost ageless, a perpetual young **man** about to launch himself on his career.

The person was perhaps better than human **beings** about keeping those pledges.

Note that two of the three chosen phrases lack of subject. This is in fact an advantage, because at the post-processing step we can freely add a subject that is related to the previously mentioned sentences.

4.5.7 Sentence Post-processing

The final sentences that will conform the stories have been already generated, but before assembling the story there are a few things that need to be checked (see Figure 13). Such things mainly concern number and gender agreement between subjects and verbs, and between quantifiers and nouns, as well as the correct use of personal, possessive and relative pronouns. Also this step checks that all transitive verbs have an assigned predicate, and all prepositions have a complement; this is done to avoid having truncated sentences and incomplete ideas. Finally, another important step that is done here is the NER substitution to avoid that recognizable entities from other stories are mentioned in ours.

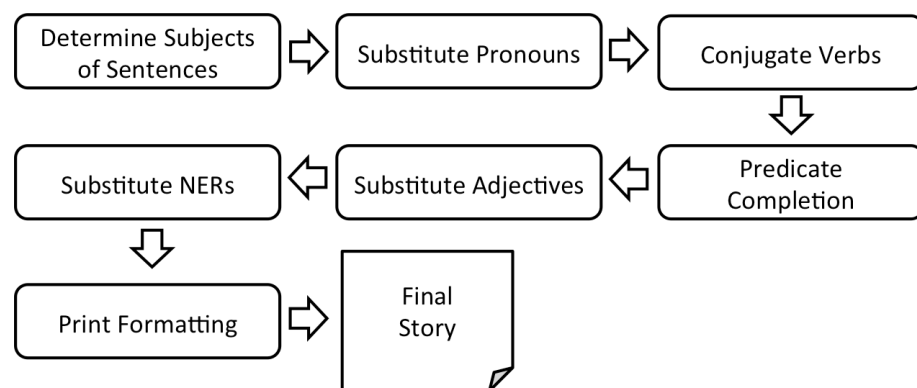


Figure 17: Post-processing module steps at the sentence level.

To enhance the explanation of this module we will describe each step with our given example. Currently, we do not have a story but a list of sentences. The purpose of this module is to improve the syntactic quality inside each sentence and to create semantic connections among all generated sentence. Our current list of sentences (in the order in which they were generated) is:

Generated Sentence

1) The white sky met the white ground in a strange **world** where the grubby black bus station floated up into a well-tended patch of blue and yellow flowering groundcover surrounding islands of taller lace-lady ferns and monarch roses.

2) Since both the **earth** and **man** are sentient beings, their emanations coincide or rather, the earth has all the emanations present in man and all the emanations that are present in all sentient beings, organic and inorganic for that matter.

3) There was a time when a strange **world** existed.

4) The shadow within the white **patch** was brighter than any star outside it.

5) The underwater **world** was so very strange, like another planet.

6-7) The **station** was a bright dot in the distance, bracketing the invisible five-space jump point. The **sky** was dawn, and the sky was changing from black to murky gray.

8-10) pawed the earth, lowered his head and drove with incredible force at the older one. was thirty odd, almost ageless, a perpetual young man about to launch himself on his career. The **person** was perhaps better than human beings about keeping those pledges.

Table 19: An example of algorithm generated sentences ready for macroplanning.

The first step of this module is to extract the subject of each sentence to ensure that every sentence has a subject. To locate the subject, first we look for the first verb that appears from left to right; then, we take the nouns that appears inside the interval of the first word and the found verb. In other words: a sentence has a subject (or many subjects) if it has a noun (or nouns) before the first verb appears from left to right. Since we know at which step was each sentence generated we can control the subjects that appear at each step. Thus, if we find there is no subject or the wrong subject, we change it. Subject extraction also includes the gender of the subject, in order to refer to it on latter sentences using the correct pronoun. At table 15 we list the subjects that should appear at each sentence. Currently in our example, the subjects of our ten sentences are: world, earth, world, patch, world, station, sky, -, -, person.

#	Target Subject	Subjects from Example	Generation Step
1)	<Main Word>	world	Main Sentence
2)	<Outcome Subj> OR <New Subject>	earth	Story Outcome
3)	<Main Word>	world	Introduction Sentence
4)	<Secondary Character>	patch	Character Description (secondary character)
5)	<Main Word>	world	Character Description (main character)
6-7)	<Active Noun per sentence> _i	station, sky	Character Description (active nouns)
8-10)	<Noun per premise> _i OR <New Subject>	-, -, person	Outcome Premises

Table 20: Subject Post-processing step

Note that almost every subject matches its target, but there is a lack of subjects in two outcome premises. Since we already used the outcome nouns on those sentences and they

are not the subjects, then we complete the sentences adding a new subject. To avoid entity incoherence, this new subject is introduced as a proper name. This new subject can be freely used in the outcome premises and the outcome itself³. The proper name generated for this occasion was *Jason*. The outcome premises are now changed to:

Jason pawed the **earth**, lowered his head and drove with incredible force at the older one

Jason was thirty odd, almost ageless, a perpetual young **man** about to launch himself on his career.

The person was perhaps better than human **beings** about keeping those pledges.

Now that we know the subject of each sentence and its respective gender the following steps are much faster: For the pronoun substitution, it checks that every pronoun mentioned inside a sentence matches the subject gender. If the subject is a neutral noun (as it happens with the majority of nouns in English) the pronouns are not changed. It also applies for subjects with proper names: if it is a feminine name, for example, then all the sentences that refer to the same subject should have a feminine pronoun.

The next step is to check for verb tense coherence. The intention is to change all verbs to past tense (except for the outcome sentence.) Also the participle verbs, gerund verbs, and the verbs preceded by “to” are left at its original tense, the rest are conjugated in simple past tense. In our example all verbs were already in past tense so the sentences were left the same.

Later there is the predicate completion where sentences that accidentally end with a transitive verb are given a predicate for that verb in order to complete the idea. Also if a sentence ends with a preposition a prepositional complement is randomly searched to complete that sentence. This step did not apply for our example. Supposing that a given sentence is “*The man told her to give*” since *give* is a transitive verb the algorithm looks for all the VPs starting with *give* and randomly assigns a predicate such as “*The man told her to give him a kiss.*”

³ If the <New Subject> was necessary at the outcome step, and it is also necessary at the outcome premises step, the subject is only generated once, and it is repeated in both steps. This is done to avoid having different subjects where only one is needed.

Afterwards, an adjective verification is done individually at each sentence. The purpose of this verification is to avoid opposed modifiers to the same nouns. This step is done with the help of the Entity Dictionary created earlier. For example, we have this two generated sentences (sentences 2 and 7) in our example:

The **white sky** met the white ground in a strange world where the grubby black bus station

The sky was dawn, and the **sky was changing from black** to murky gray.

Sentences 2 and 7 need to have the same subject (with the same characteristics), which is sky; since both sentences are produced based on the same noun, it is impossible for them to have antonyms (even if the adjective is not directly modifying them), so the word *black* is changed to the original word that provoked the antonym: *white*. Based on this, our sentence changes to:

The sky was dawn, and the sky was changing from **white** to murky gray.

At last, the NER substitution is done. On Section 3.5.5 we listed the nine different Named Entity Categories that NLTK manages. There are many cases where the extracted sentences will contain those entities, which are substituted by internal tags in our Knowledge Base. At this step the task is to properly substitute those tags (See table 16) with a coherent subject or in other cases with an ambiguous word in order to avoid recognizable entities and keep a sense of vagueness on the story.

NER Tag	NER Substitution
<<PERSON>>	<Character Name>
<<ORGANIZATION>>	someone
<<LOCATION>>	somewhere
<<DATE>>	yesterday
<<TIME>>	sometime
<<MONEY>>	money
<<PERCENT>>	100%
<<GPE>>	someplace

Table 21: NER substitution in the Post-processing module

The last step of post-processing is the visual formatting and correct sentence ordering to present the final story to the user. We explain this step on the next section.

4.5.8 Final Story

We have established the linguistic generation algorithm as a bottom-up approach; this means that linguistic production inside phrases is done almost independently from one another as shown earlier, but at the final step we still follow a top-down method that is in charge of linking phrases coherently, so the actions or descriptions inside stories seem to advance the plot. This is the step that acts as the macroplanning module, which runs together with the story post-processing module to produce the definitive output.

In order to cover even more the rigid macroplanning structure (the macroplanning is a series of slots called sentence labels, showed on Table 17), the number of sentences that the final output contains is variable: it depends on the number of nouns that both the main sentence and chosen story outcome contain. Every story starts with an introduction sentence that presents the character, followed by the chosen main sentence. If the main character is a noun that can take a name (an animal, a human being, an animated entity in general...) a sentence will introduce his/her name. The story continues describing a secondary character (if another active noun is found, if not, it is omitted). Next, a descriptive phrase of the main character is provided to give the reader more information about him/her. In addition, some phrases are provided to introduce and/or describe the active nouns that will appear on the story outcome (outcome premises); if the chosen outcome does not contain an active noun these sentences are omitted. Finally, the story outcome is mentioned and the story always ends with this kind of sentence (a complex clause.)

Sentence Label	Generated Sentence
Introduction Sentence	There was a time when a strange world existed.
Main Sentence	The white sky met the white ground in a strange world where the grubby black bus station floated up into a well-tended patch of blue and yellow flowering groundcover surrounding islands of taller lace-lady ferns and monarch roses.
Main Character Name (if applicable)	[Not applicable]
Most Connected Entity Description (Secondary Character)	The shadow within the white patch was brighter than any star outside it.
Main Character Description	The underwater world was so very strange, like another planet.
Active Nouns Descriptions	The station was a bright dot in the distance, bracketing the invisible five-space jump point. The sky was dawn, and the sky was changing from white to murky gray.
Outcome Premises	<p>Jason pawed the earth, lowered his head and drove with incredible force at the older one.</p> <p>Jason was thirty odd, almost ageless, a perpetual young man about to launch himself on his career.</p> <p>The person was perhaps better than human beings about keeping those pledges.</p>
Story Outcome	Since both the earth and man are sentient beings, their emanations coincide or rather, the earth has all the emanations present in man and all the emanations that are present in all sentient beings, organic and inorganic for that matter.

Table 22: Macroplanning Structure

Besides correct sentence ordering, this step handles punctuation among sentences, it verifies that quotes open and close correctly and normalizes a single space between words. Especially with the outcome premises and active noun descriptions, since they are complex constructed sentences the punctuation verification is important. Once that everything is checked the final story is produced. We can see our full final example:

There was a time when a strange world existed. The white sky met the white ground in a strange world where the grubby black bus station floated up into a well-tended patch of blue and yellow flowering groundcover surrounding islands of taller lace-lady ferns and monarch roses.

The shadow within the white patch was brighter than any star outside it. The underwater world was so very strange, like another planet. The station was a bright dot in the distance, bracketing the invisible five-space jump point. The sky was dawn, and the sky was changing from white to murky gray. Jason pawed the earth, lowered his head and drove with incredible force at the older one; Jason was thirty odd, almost ageless, a perpetual young man about to launch himself on his career. The person was perhaps better than human beings about keeping those pledges.

Since both the earth and man are sentient beings, their emanations coincide or rather, the earth has all the emanations present in man and all the emanations that are present in all sentient beings, organic and inorganic for that matter.

5. Evaluation and Results

In this chapter we will explain how do we evaluate the performance of our story generator. First we give details about the corpus used to construct the knowledge base. We continue by describing the parameters that were taken into account to generate better quality stories. We decided to evaluate the generated stories based on a survey where people were asked to score different parameters in texts; so next, we will establish the aspects that are measured in the evaluation, and also the restrictions. Finally, we explain how the survey was applied to human evaluators and the obtained results.

5.1 Corpus

The experimental setup was done with a corpus of 9,560 books. They were mostly science fiction novels and short stories, as well as famous works in literature. The generator was trained on that corpus, resulting on more than 2 million processed sentences and more than 350,000 unique tokens (a single word could be converted into three tokens: the word-as-adjective, word-as-verb and word-as-noun was counted separately).

The processing resulted in the following literary structures:

Structure Type	Number of Tokens
NPV	3,329,819
VPP	1,535,086
PPP	1,397,435
PHR	1,391,762
CLS	59,861

Table 23: Number of literary structures found in the experimental corpus.

5.2 Story Selection

We decided to generate stories based on the words that were more frequent on our corpus in order to maximize the *creative* options for the algorithm: the more frequent the word, the more structures can be used to combine among them. We automatically extracted a total of 300 words: the top one hundred nouns, top one hundred verbs, and top one hundred adjectives. Once we had this list, the *favorite* thirty words (ten of each type) were chosen based on the number of connections that those words held in the knowledge base.

Finally, we generated five instances of stories based on each of those thirty words, giving a total of 150 stories. In the end, we only needed to include four stories in our survey, so four of the *best* stories were chosen among the 150 to be placed on the survey.

5.3 Evaluation Survey

The survey is inspired on the Turing test, where people have to decide if the text they are reading was produced by another person or by an algorithm. Additionally, we decided to measure four more parameters: coherence, interest, originality and syntax quality. It was applied via webpage to encourage anonymity and to obtain more evaluators.

5.3.1 Survey Design

The survey included seven texts: three were taken from human authors and four were from our algorithm (See table 18), they were only identified by a neutral ID so the evaluator could not know which texts were human and which were artificial.

ID	Title	Author	# of Words
Text 1	Naked Lunch (fragment)	William Burroughs	216
Text 2	A Big Man Existed	Machine	124
Text 3	The Left Gets Threatened	Rachel Summer	174
Text 4	The White Sky Met	Machine	175
Text 5	A Beautiful Story	Machine	218
Text 6	Finnegans Wake (fragment)	James Joyce	122
Text 7	The First Man	Machine	234

Table 24: The texts that were presented to human evaluators

There are five questions for each text, and the evaluator can answer only by using a slider bar with two antonyms on the extremes, where she has to decide how close is her opinion to a given concept (See figure 16). This type of questionnaire is inspired on Charles Osgood's semantic differential. Osgood designed a questionnaire where people's opinions and tendencies towards opposite concepts are measured in search of an intended meaning (Osgood, Suci, & Tannenbaum, 1957). Since word meanings are not easily measured, we benefited from this design and by letting people show their tendencies towards the answers, instead of forcing them to decide on a binary question, we obtained a distribution of the opinions towards the desired concepts.

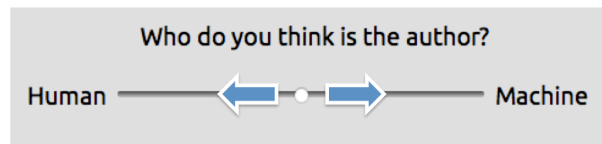


Figure 18: The slider allows to show a tendency towards a chosen concept

We can divide the survey in five different sections, one for each type of question or parameter. Here is an example of how a question looks on the actual survey:

Text 5

Once upon a time there was a beautiful story and a narrator. The narrator heard Zoe's beautiful story of the life of a person, born a slave but devoted himself for the first and – so far – last time. The person had played his part. The Wise One was beautiful, with her long red-gold hair and clear green eyes. The narrator aimed her recorder at adapted skips, full of grain and vegetables being hauled by domesticated six-packs.

She had been a noted philosopher, in her day, among many other things. The Wise One herself was carrying the small satchel, the original sin of which we hear so much was, in all truth, this same materialism.

The priestly response of secrecy had been flawed from the beginning. Call it a trampled national pride, call it cynical realism; the warm, friendly nature of my fellow man has kept me pretty much constantly on the move.

Time was to sin in secrecy, to indulge in that sloth and pride, to covet the unlawful, to yield to the promptings of your lower nature, to live like the beasts of the field, nay worse than the beasts of the field, for they, at least, are but brutes and have no reason to guide them: time was, but time shall be no more.

How coherent does this text seem to you?

Incoherent ————— Coherent

How creative is it?

Original ————— Common

How interesting is it?

Boring ————— Interesting

Who do you think is the author?

Human ————— Machine

Consider its general quality:

Bad English ————— Good English

Figure 19: Each question has five different parameters that are separately evaluated

As it can be seen on Figure 17, the five questions consist on a slider between two opposite concepts. An evaluator is asked to give her preference towards a given concept. Each slider has a hidden value between 0.00 (the left most part of the slider) and 10.00 (the right most part of the slider), and it starts with the default *neutral* value of 5.00. The complete table of aspects and concepts is shown on Table 19. It is important to emphasize that sometimes the *positive* concept of an aspect is situated on the right and sometimes it is on the left of the slider; this was done to avoid visual tendencies on the survey.

Aspect	Concept A (value < 5)	Concept B (value >= 5)
Coherence	Incoherent	Coherent
Creativity	Original	Common
Interest	Boring	Interesting
Authorship	Human	Machine
Syntax	Bad English	Good English

Table 25: Aspects and concepts considered on the proposed survey

It is also important to highlight that evaluators were never explicitly informed about the possibility that a text was artificially generated (although we are aware that they should deduce it by themselves from the questions); they were only asked to answer a survey about different styles inside fiction texts.

5.3.2 Chosen Texts

As we said earlier, the survey consists of seven texts, three *human* (we will successively call *human texts* the ones that were completely written by a known author) and four *artificial* (we refer to our algorithm’s generated texts as *artificial texts*, since there is not a common source where the complete text can be found as it is presented here). The three human texts were chosen because of their postmodern characteristics; as we discussed on Section 3.3.3, postmodern authors (and their texts) are distinguished because of their intention of exploring language. Since our algorithm is an automatic attempt to explore language, we decided to contrast our artificial explorations with similar human experiments in order to be fair.

First we will explain why we chose the human texts. The first chosen text is a fragment of *Naked Lunch*, a novel from William Burroughs, which is mainly famous because of its obscene language and situations, as well as its intentional semantic incoherence, and of course its challenge to American culture. Here we present the fragment that was included on the survey:

I can feel the heat closing in, feel them out there making their moves, setting up their devil doll stool pigeons, crooning over my spoon and dropper I throw away at Washington Square Station, vault a turnstile and two flights down the iron stairs, catch an uptown A train..Young, good looking, crew cut, Ivy League, advertising exec type fruit holds the door back for me. I am evidently his idea of a character. You know the type: comes on with bartenders and cab drivers, talking about right hooks and the Dodgers, calls the counterman in Nedick's by his first name. A real asshole. And right on time this narcotics dick in a white trench coat (imagine tailing somebody in a white trench coat. Trying to pass as a fag I guess) hit the platform. I can hear the way he would say it holding my outfit in his left hand, right hand on his piece: "I think you dropped something, fella."

But the subway is moving.

"So long flatfoot!" I yell, giving the fruit his B production. I look into the fruit's eyes, take in the white teeth, the Florida tan, the two hundred dollar sharkskin suit, the button-down Brooks Brothers shirt and carrying The News as a prop. "Only thing I read is Little Abner."

The second text is a contemporary story named *The Left Gets Threatened*. It was chosen because its syntactic characteristics are very similar to our artificial stories; and also because it starts with the classic phrase "Once upon a time..." Besides, it has a semantic challenging character: a Left shoe; it also is a short story made of just 174 words; and finally, it does not have an explicit ending, meaning that the reader is intended to infer it from the context (another characteristic of our artificial texts!) The complete story goes as follows:

Once upon a time a famous brand of sneaker with tie-up laces and a zipper on the heel decided to walk somewhere with no feet inside it. It was a Left.

It was independent, that Left. It didn't need direction or a mate or even a destination. Out it went, wandering first to the part of the neighborhood where the goody-goody kids lived and then down by the Marina. A gorgeous, sunny day. The Left was in bliss.

"Hey, Left!" came a sudden unwelcome voice. It was Maurice, who never left his gang far behind. "What do ya think you're doing out here all alone," Maurice continued, sounding slimy as usual. The others in the gang circled the Left gangishly, salivating.

Just then Reesa came hopping over, a friend from the Star Trek expos occasionally held downtown. Her piercing green eyes fried the roving gang; their smoking ashes smelled of rot, dirty rag, and a hint of cinnamon.

"Thanks, Reesa," said the Left, before hopping, embarrassed, into the water, never to be seen again.

The last text from an actual author is from James Joyce: a fragment of *Finnegans Wake*. It is considered by many as one of the most difficult novels ever written in English language. Naturally, we chose it because of its syntactic, grammatical and lexical variations; every sentence is abundant in ironic and critical sub textual critics to language from the author, as well as almost endless neologisms and previously untold expressions. Because of its immense complexity we only could choose a (more less *coherent*) fragment of the novel. The chosen fragment was:

It was of a night, late, lang time ago, in an auldstane eld, when Adam was delving and his madameen spinning watersilts, when mulk mountynotty mas was everybully and the first leal ribberrobber that ever had her ainway everybuddy to his lovesaking eyes and everybilly lived alove with everybiddy else, and Jarl van Hooter had his burnt head high up his lamphouse, laying cold hands on himself. And his two little jiminies, cousins of ourn, Tristofer and Hilary, were kickaheeling their dummy on the oil flure of his homerigh, castle and earthenhouse. And, be dermot, who come to the keep of his inn only the niece-of-his-in-law, the prankquean. And the prankquean pulled a rosy one and made her wit foreninst the dour.

On the other hand, we chose our artificial stories for the survey, as we explained on Section 5.2, based entirely on our intuition and on our intention to mix the syntactic and semantic characteristics with the human texts.

The first text is about a big man named Jacob. It was generated after the input word *man*. One of the particular characteristics of this text is the presence of a quote, a character speaking, which could be confused syntactically with the story of William Burroughs and Rachel Summer: we wanted to demonstrate that our algorithm can also include a coherent quote in the story. The complete text is:

There was a time when a big man existed. The big man nodded, raced around ordering that they needed to update the old plans: the original strategy of mass assault. His name was Jacob. The assault had not been a complete success. Jacob was encrusted with sand.

They were relatively small of human size in fact and within it they could see more of the glowing shadows. Jacob was big, grey-haired, athletic-looking, more deliberate in his motions. The criminal world is a totally unbelievable, blood soaked, insane, comedy of the sitter 's face.

"Any man in the world listens to a bell that rings for any reason whatsoever," Jacob said, affecting a schoolteacherish tone, "and no man can possibly avoid gawking at a fire."

The next text was generated based on the noun *world*. It is actually the example that we used in Chapter 4 to describe step by step the story generation process. We made a little change on the story presented in the survey: We subtracted the introduction sentence. This was done because it gave a more dramatic beginning to the story, and also with the purpose of avoiding the reader to realize that all our artificial stories have an introduction sentence. The complete text presented on the survey was:

The white sky met the white ground in a strange world where the grubby black bus station floated up into a well-tended patch of blue and yellow flowering groundcover surrounding islands of taller lace-lady ferns and monarch roses.

The shadow within the white patch was brighter than any star outside it. The underwater world was so very strange, like another planet. The station

was a bright dot in the distance, bracketing the invisible five-space jump point. The sky was dawn, and the sky was changing from white to murky gray. Jason pawed the earth, lowered his head and drove with incredible force at the older one; Jason was thirty odd, almost ageless, a perpetual young man about to launch himself on his career. The person was perhaps better than human beings about keeping those pledges.

Since both the earth and man are sentient beings, their emanations coincide or rather, the earth has all the emanations present in man and all the emanations that are present in all sentient beings, organic and inorganic for that matter.

The third artificial text was generated based on the word *beautiful*. It goes as follows:

Once upon a time there was a beautiful story and a narrator. The narrator heard Zoe's beautiful story of the life of a person, born a slave but devoted himself for the first and -- so far -- last time. The person had played his part. The Wise One was beautiful, with her long red-gold hair and clear green eyes. The narrator aimed her recorder at adapted skips, full of grain and vegetables being hauled by domesticated six-packs.

She had been a noted philosopher, in her day, among many other things. The Wise One herself was carrying the small satchel, the original sin of which we hear so much was, in all truth, this same materialism.

The priestly response of secrecy had been flawed from the beginning. Call it a trampled national pride, call it cynical realism; the warm, friendly nature of my fellow man has kept me pretty much constantly on the move.

Time was to sin in secrecy, to indulge in that sloth and pride, to covet the unlawful, to yield to the promptings of your lower nature, to live like the beasts of the field, nay worse than the beasts of the field, for they, at least, are but brutes and have no reason to guide them: time was, but time shall be no more.

Lastly, a text generated based on the adjective *lazy*. Here is the complete text:

Once upon a time there was a first man. The first man, yawning, sleepy and bleary-eyed, the lazy beast, stumbled along linguistic pathways that were something other than the most direct ones. His name was Owen. The great ones had gone to discuss high matters. Owen had brought shame to his door. The beast was not a showy beast, and it was rather small, having much of the blood.

The fact that this poor simple mental retard couldn't make it work is beside the point. The only reason the person would be going into the water was to do something nasty with the nuclear mines, as the mundane world saw no great profit, commercial or artistic, to be reaped from our little field; a vague, unsatisfactory basis on which to risk the only life. The demon, steel strong and more than iron hard, leaped free to dispose of the men before him and around him. In somewhere, the rainforests of the people are being destroyed at an alarming rate by bulldozing and burning.

It might have a real basis in fact, too, but the real reason is that we feel that a world with tigers and orangutans and rainforests and even small unobtrusive snails in it is a more healthy and interesting world for humans (and, of course, the tigers and orangutans and snails) and that a world without them would be dangerous territory.

5.3.3 Evaluators

The survey was successfully applied to a population of 32 individuals, all of them students. We tried to keep a balance in a number of factors in order to reduce some of the ambient variables that are always present when dealing with human evaluators. We kept track of gender and age balance. A fraction of our evaluators is still studying a degree, but most of the population has already finished a bachelor's degree, some of them are now employees and others are studying a postgraduate. The complete distribution is show here:

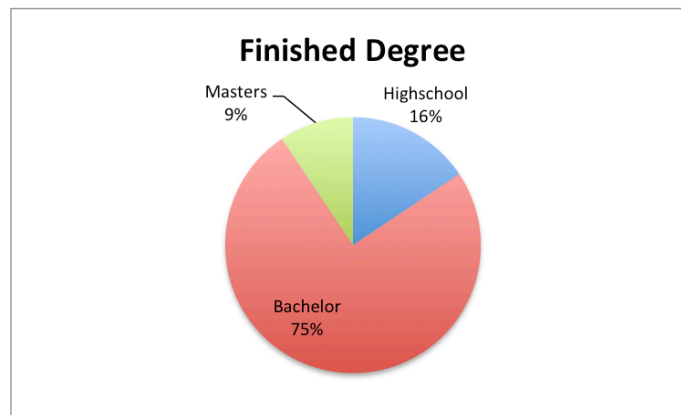


Figure 20: Distribution of population by degree level.

Also the field of study of every evaluator was relevant. We managed to balance evaluators from heterogeneous fields such as humanities (e.g. philosophy, linguistics, literature), engineering (e.g. biotechnology, computer, electronics), and sciences (e.g. cognitive

neuroscience, math, chemistry). The complete population distribution by field of study is presented on the next chart:

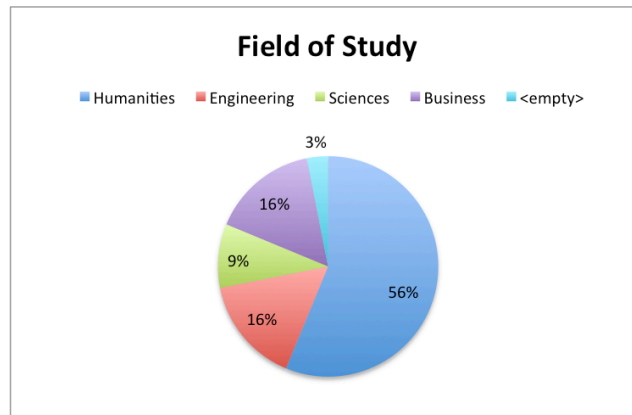


Figure 21: Distribution among evaluators' fields of study

Finally, we considered important to selected individuals that had moderate to strong reading habits. The distribution of reading habits is shown, on books per year, in the next chart:

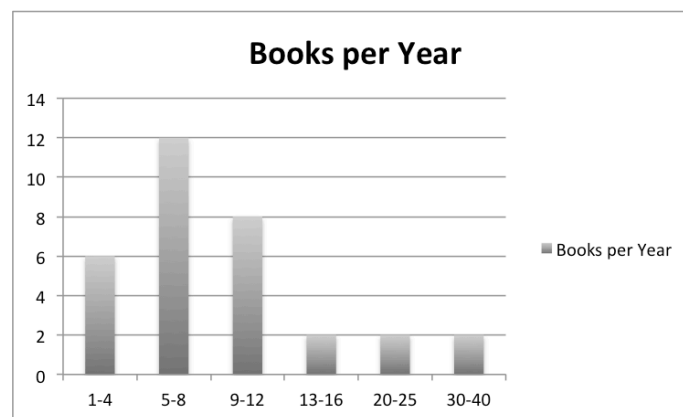


Figure 22: Distribution of evaluators' reading habits

Now that we have stated how the survey was designed, and the parameters that will be measured, as well as the characteristics of the evaluators we will present the results.

5.4 Results

We will look at the results from the two main perspectives that were stated on our objectives: to produce texts that could be considered creative but still hold semantic coherence; and the other aspect that is highly relevant: to identify where do our texts situate in respect to existent postmodern works in literature (*human* texts). We will observe that our story generator fulfilled our expectations.

We split in two our survey results, that is, we evaluated separately the scores given to human texts and the scores given to artificial texts. Figure 23 and Figure 24 show the comparison of the number of votes obtained by each kind of text. The votes for human texts were counted using the average of the three human texts present in the survey. Like shown in Table 25, if a value was less than five it was considered a vote for the left slider concept, likewise if a value is more equal than 5 it was considered as a vote for the right concept. It worked the same way with the average of Artificial Texts (the votes were counted using the average of results from the four texts present in the survey.) First, we compare how were they evaluated in terms of the positive concepts:

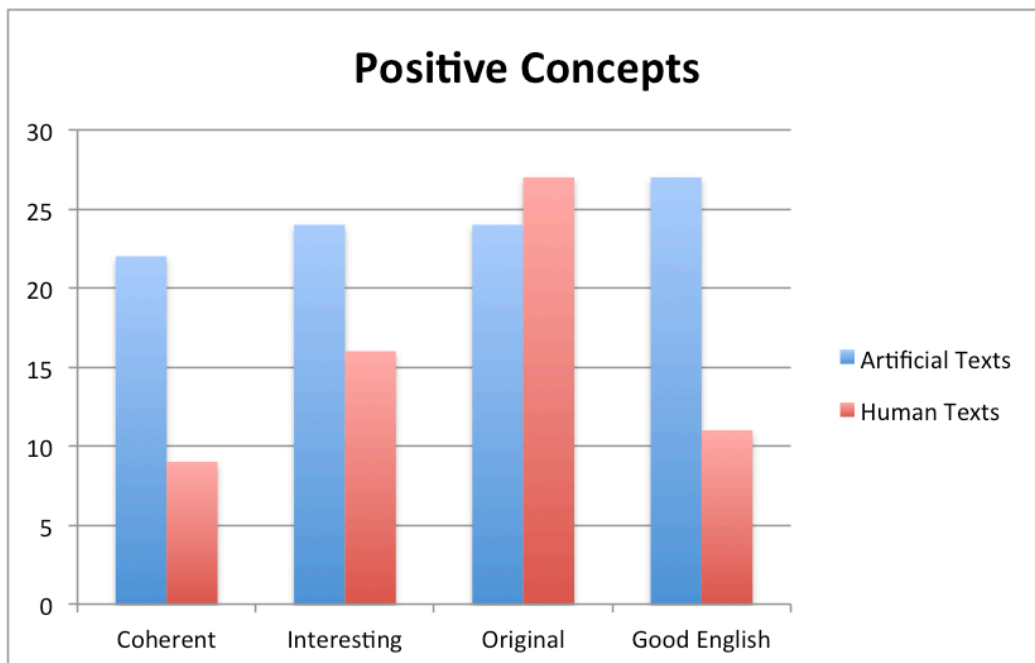


Figure 23: Positive votes received for positive concepts of each aspect.

From the artificial texts, it can be easily deduced that they succeeded in every evaluated aspect. The average of artificial texts from our survey obtained more votes than the average of human texts on almost every positive aspect (except originality, but still it received a high voting rate). From Figure 21 we can infer that evaluators considered our texts as coherent, interesting, original and with good English.

Human texts, on the other hand, did pretty low in terms of syntax and coherence. This is understandable, since the chosen texts do not hold a common use of language and they also are part of bigger works (the exception is Text 3: the short story by Rachel Summer), so maybe coherence would have been better if the evaluators had read the whole novels from where these texts were extracted.

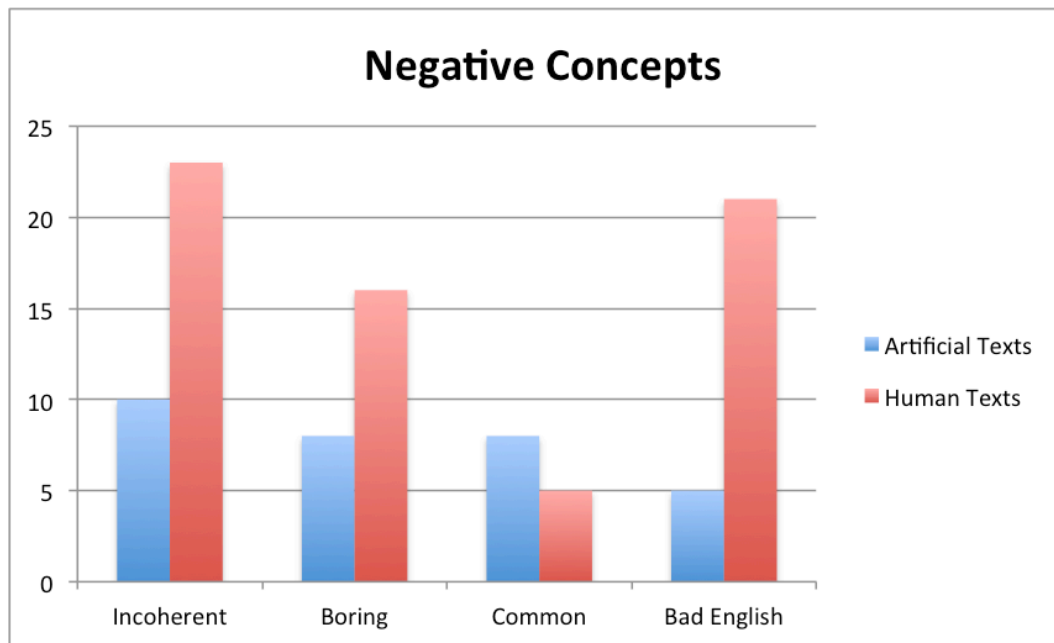


Figure 24: Votes received for the negative concepts of each aspect.

The same comparison was done considering the *negative* concepts of each aspect. In this case, the aim is to minimize the votes for every concept. Figure 22 shows that artificial texts receive less negative votes than human texts. The creativity aspect, considering the last two figures confirms that our texts, both human and artificial, were correctly chosen, since the votes for originality are a lot more than the votes for commonness. This only

reinforces our argument claiming that our texts are capable of emulating creativity in the same way as human texts do.

To stress even more the creativity perceived on our texts, we include a more detailed graph showing the percentage of votes that human and artificial texts obtained in each interval (we divided the slider in ten intervals to measure this).

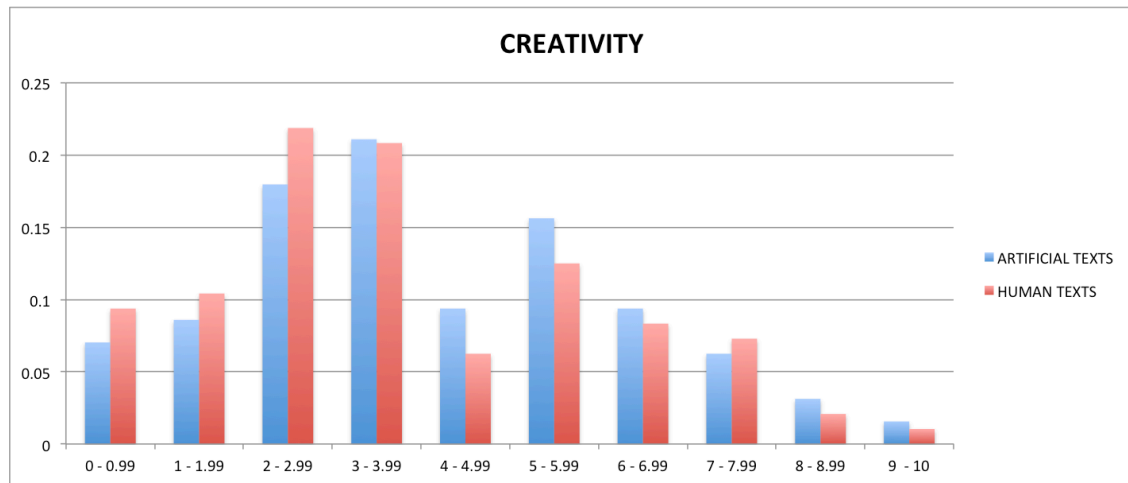


Figure 25: Percentage of votes obtained in the creativity aspect of texts.

We recall that the intervals closer to zero mean originality and the ones closer to ten mean that the texts are common. It is clearer in Figure 25 the fact that all texts included in the survey have creative characteristics (whatever those may be), since both artificial and human texts have the higher percentages of votes inclined towards the intervals that are closer to zero.

There still is one more general chart that shows the overall semantic differential, which is an attempt to measure concept meanings by asking the evaluator where her position towards two opposite adjectives lie (Osgood, Suci, & Tannenbaum, 1957). In this case we measured the overall perception from evaluators on texts (see Figure 26.) This chart shows the semantic categorization of our generated texts, concluding that in general our texts were taken with a positive judgment. We included in the same chart the semantic differential of human texts, in order to emphasize even more the good results that our algorithm obtained.

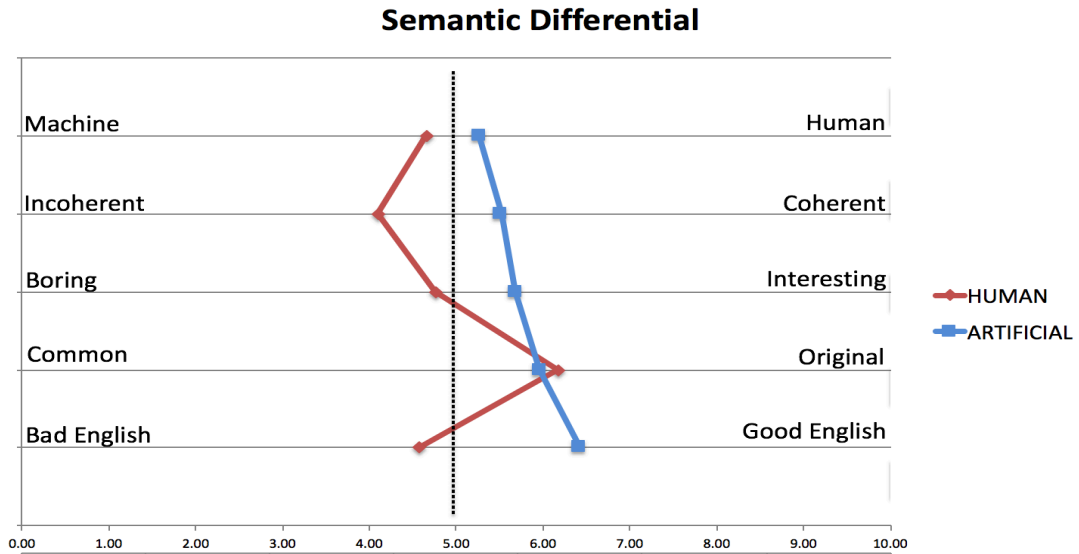


Figure 26: Total semantic differential based on the total population responses.

On the whole, data show integration among generated texts and human postmodern texts. Our objectives aimed to avoid rigid text generation, explore new meanings in language and mimic postmodern writing. Results show that our text generator meets the desired goals. On the next chapter we will explore more profoundly the conclusions that can be drawn from our survey.

6. Conclusions

We successfully achieved the specific objectives of this thesis: relevant word relations were extracted from a constructed corpus of fiction. We also managed to extract syntactic patterns and ensemble them together in different ways allowing meaning to emerge from the diverse combination of words. We proposed an architecture that is able to produce novel sentences based on what was previously written and also manages to create a narrative flow in texts resulting in stories that hold a literary style and give the impression of creativity.

Our evaluation demonstrates that on the whole we achieved the general objective proposed by avoiding an over structured text and generating coherent stories with novel sentences based on what was previously written and also by creating a narrative flow in texts resulting in coherent stories.

6.1 Contributions

The main contribution of this thesis is the proposal of a new architecture that combines diverse fields such as literary theory, stylistics and text generation. The proposed framework is an entire methodology that converts a single word into a whole story, by using syntactic and semantic characteristics of words and phrases, thus building a bridge between language syntax and semantics. We also proposed, based on the JCN similarity measure, a new semantic measure between two sentences in order to know which sentence is semantically closer to another sentence by analyzing its word content.

Besides, unlike the majority of previous generation frameworks, we decided to give more importance to style than structure. Furthermore, we achieved the impression of creativity because meaning is also an inherent property of style, meaning is not merely inside word definitions and sentence content, but in the manner in which those words and sentences are combined. For this reason, our texts managed to camouflage among human authors: we succeeded in the imitation of postmodern style.

Nevertheless, we need to be cautious about our results. We specifically built texts based on postmodern literature theory. We rely on the assumption that most of the meaning that a text holds is created on the reader's mind. By doing this, we avoid the problem of producing texts with intentionality, and focused on language experiments and guided word combinatory.

6.2 Future Work

Since we proposed an entire methodology for fiction text generation, we identified various points that can improve to obtain better stories. The most obvious is the need of a more flexible macroplanning algorithm; currently generation focuses on content variations at sentence level, but always holds the same story structure. It would be interesting to apply our microplanning ideas into a more powerful story planner able to improve story flow and create different kinds of narrative structures.

The phrase ranking algorithm is also an area that can be improved with more and better rules. Currently we proposed a small set of rules, both semantic and syntactic, that allowed us to have control of sentence coherence; however, we believe that a larger set of rules could and should be implemented to improve sentence quality and combination possibilities. One of the biggest areas of opportunity is on subordinates, which we successfully identified (and categorized in four classes), but we were not able to take advantage of their content because of the lack of proper ranking rules to avoid generating incoherent sentences.

Our approach also looks specifically for text verbosity. This can be seen specially on the Main Sentence generation step and the Word-Clause Similarity measure. We strongly favor phrases that have more words. This approach fulfilled our expectations; nonetheless, we would like to experiment with different parameters that allow us to control text length, and as a result, be able to produce short or long stories depending on specific requirements.

Finally, one of the biggest problems we encountered when processing our corpus was the Named Entity Recognizer, because the most common NERs were trained on news corpora mainly, and we would have highly benefited from a trained NER on fiction texts. We believe that creating a specific NER model for fiction corpora would strongly benefit future works in this area.

On the whole, the state of the art is still far from imitating different kinds of literature, other than postmodern texts, that require more elaborated knowledge and experience of *Reality*. We still can safely say that literature is a field dominated by humans, where sentences and text go beyond syntax and hold complex semantic and pragmatic relations, which are only produced and validated by conscious beings. On the other hand, we think that this is not an obstacle for computer science and natural language processing to explore new ways to automatically create text and meaning. In fact, we believe that fiction texts are excellent material to explore the complexity of human knowledge in general.

We also think that creativity is still an underestimated field in terms of what can be achieved automatically. On our definition of creativity, we emphasized the parallelism that exists between some AI concepts (such as search space, process, knowledge and goals) and the creativity concepts. We must highlight the view of creativity as new combinations of what already exists. It is a fact that machines are much better than humans at searching and processing enormous quantities of data; and since creativity is very close to the concept of combinatory, it might be possible to develop new frameworks that formalize it, not only in art but in any human activity that can be called creative.

References

- Abrams, M. H. (1953). *Tradition, The Mirror and the Lamp: Romantic theory and the Critical*. Oxford University Press.
- Alba, E., & Dorronsoro, B. (2008). *Cellular Genetic Algorithms*. New York: Springer.
- Argamon, S., & Burns, K. D. (2010). *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*. Springer-Verlag Berlin Heidelberg.
- Burroughs, W. (1959). *Naked Lunch*. New York: Grove Press.
- Barbieri, G., Pachet, F., Roy, P., & Esposti, M. D. (2012). Markov Constraints for Generating Lyrics with Style. *European Coordination Committee for Artificial Intelligence (ECCAI)* , 115-121.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. Sebastopol: O'Reilly Media Inc.
- Black, J., & Wilensky, R. (1979). An Evaluation of Story Grammars. *Cognitive Science* , 213-229.
- Blair, D. (2006). *Wittgenstein, Language and Information: "Back to the Rough Ground!" (Information Science and Knowledge Management)*. Springer.
- Borges, J. L. (1944). *Ficciones*. Madrid: Anagrama.
- Calvo, H. (2013). *Procesamiento práctico de Lenguaje Natural*. México: Editorial Sociedad Mexicana de Inteligencia Artificial A.C.
- Cohen, A. (2010). Driving the Creative Machine. *Orcas Center, Crossroads Lecture Series*.
- Eco, U. (1979). *Lector in Fabula: La cooperación interpretativa en el texto narrativo*. Barcelona: Lumen.
- Derrida, J. (1967). *De la gramatología*. Mexico: Siglo XXI.

- Dinu, G., & Baroni, M. (2014). How to make words with vectors: Phrase generation in distributional semantics. *Proceedings of the 52nd Annual Meeting of the ACL*.
- Gervás, P. (2013). Propp's Morphology of the Folk Tale as a Grammar for Generation. *Computational Models of Narrative*. Hamburg.
- Grice, H. P. (1975). Logic and Conversation. In Cole, *Syntax and semantics 3: Speech arts* (pp. 41-58). London: Elsevier.
- Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing*. Prentice Hall.
- Jiang, J. J., & Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *Proceedings of International Conference Research on Computational Linguistics* , 15.
- Joyce, J. (1939). *Finnegans Wake*. London: Faber & Faber.
- Kripke, S. (1980). Naming and Necessity. USA: Harvard University Press.
- Lavoie, B., & Rambow, O. (1997). A fast and portable realizer for text generation systems. *Proceedings of the 5th ANCL*, (pp. 265-268). Washington.
- Lakoff, G. (1972). Structural complexity in fairy tales. *The Study of Man* , 128-150.
- Marina, J. A. (1993). *Teoría de la inteligencia creadora*. Barcelona: Anagrama.
- McIntyre, N., & Lapata, M. (2009). Learning to Tell Tales: A Data-driven Approach to Story Generation. *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics* .
- McKee, R. (1997). *Story: Substance, Structure, Style and the Principles of Screenwriting*. New York: Harper-Collins Publishers.
- Meehan, J. R. (1977). TALE-SPIN an Interactive Program that writes Stories. *Proceedings of the 5th International Conference on Artificial Intelligence* .
- Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM* , 38 (11), 39-41.

- Montfort, N., & Fedorova, N. (2012). Small-Scale Systems and Computational Creativity. *International Conference on Computational Creativity* , 5.
- Payne, T. E. (2011). *Understanding English Grammar: a linguistic introduction*. New York: Cambridge University Press.
- Pérez y Pérez, R. (1999). MEXICA: A Computer Model of Creativity in Writing. *DPhil Dissertation* .
- Pinker, S. (2014). *The Sense of Style*. New York: Penguin Group.
- Propp, V. (1968). *Morphology of the Folktale*. Austin: American Folklore Society & Indiana University.
- Swartjes, I., & Theune, M. (2006). A Fabula Model for Emergent Narrative. (Springer, Ed.) *Technologies for Interactive Digital Storytelling and Entertainment Conference* , 49-60.
- Schank, R. (1992). The Mechanics of Creativity. In R. Kurzweil, *The Age of Intelligent Machines*. MIT Press.
- Reiter, E., & Williams, S. (2010). Generating Texts in Different Styles. In S. Argamon, & K. D. Burns, *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning* (pp. 59-75). Springer-Verlag Heidelberg.
- Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems*. New York: Cambridge University Press.
- Turing, A. (1950). Computing Machinery and Intelligence. *Mind* , 59, 433-460.
- Turner, S. (1994). *The creative process: A computer model of storytelling and creativity*. Hillsdale, New Jersey.
- Wittgenstein, L. (1953). *Philosophical Investigations*. Oxford: Blackwell Publishing Ltd.
- Wittgenstein, L. (1921). *Tractatus Logico-Philosophicus*. Madrid: Alianza.