Automatic Text Generation by Learning from Literary Structures

Angel Daza & Hiram Calvo & Jesús Figueroa-Nazuno
Centro de Investigación en Computación, Instituto Politécnico Nacional, CIC-IPN
J. D. Bátiz e/ M. O. de Mendizábal, 07738, Mexico City, Mexico
{josdaza.a@gmail.com,hcalvo@cic.ipn.mx,jfn@cic.ipn.mx}

Abstract

Most of the work dealing with automatic story production is based on a generic architecture for text generation; however, the resulting stories still lack a style that can be called literary. We believe that in order to generate automatically stories that could be compared with those by human authors, a specific methodology for fiction text generation should be defined. We also believe that it is essential for a story to convey the effect of originality to the person who is reading it. Our methodology proposes corpus-based generation of stories that could be called creative and also have a style similar to human fiction texts. We also show how these stories have plausible syntax and coherence, and are perceived as interesting by human evaluators.

1 Introduction

Natural Language Generation (NLG) is the process of constructing natural language outputs from non-linguistic inputs: its task is to map meaning to text. The task of automatically generating human language has proved to be much more difficult than it had been expected. In general, concepts strongly tied to human intelligence, such as art, creativity and storytelling, are just beginning to be seriously explored with an automatic approach. That is because of the problems they trigger, such as dealing with meaning, intentionality, planning and common sense knowledge, just to mention a few.

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

We think that current approaches do not attend to the fact that literature, unlike other kinds of texts such as news or business reports, does not aim to solely inform about raw facts on some information domain of the world. On the contrary, the primary intention of literary texts is to use language properties to describe events aesthetically, with a certain special *style*, and through these descriptions transform the way we perceive reality (Eco, 2005). So, what we call literary style is the act of producing meaning not just with content, but with the uses of language and text structure. A literary text is written to find new ways of expressing reality, and in general, to seek new meanings in language use. What we intend is not only to produce structured logical texts, but to create an architecture that can produce stories that pursue the imitation of the properties that acknowledged here as *literary*.

We can distinguish three main current approaches to story generation: problem-solving, story grammars and the corpus-based approach. All of them focus on adapting existing universal generation techniques to the formalization of storytelling. This means that they tackle automatic generation of text by implementing story structures as a planning task instead of a linguistic problem, leaving behind the fact that language in literature is used in a more strongly distinct way than in everyday life.

The problem-solving approach sees generation as a search task, where a path needs to be found between a starting point and a goal state, and tries to implement common AI techniques to attack this problem. Examples of this approach are in (Meehan, 1977; Swartjes and Theune, 2006; Pérez, 1999).

Proceedings of the Fifth Workshop on Computational Linguistics for Literature, NAACL-HLT 2016, pages 9–19,
San Diego, California, June 1, 2016.

The problem with that kind of work is that, while it tends to produce coherent stories, it also tends to be limited in the kind of stories generated, primarily because it mostly relies on hand-made factual databases.

Story grammars are also a common way to approach generation. It tries to formalize storytelling theory into generative grammars that produce text with a defined structure that can be called a story. Examples in this area are in (Thorndyke, 1977; Gervás, 2013). Like the problem-solving approach, the main weakness of story grammars is the high dependence on given information, in this case handmade rules and predefined story-grammar structures.

More recently, the corpus-based approach has been used to eliminate human intervention – as much as possible – from the process of creating a story. The most successful work that follows this approach is (McIntyre and Lapata, 2009), where almost no human intervention is needed, but the results, while coherent, do not resemble a style that is easily recognizable as literary.

Here we present an example of a story produced by this approach:

> *The family has the baby. The baby is to seat the lady at the back. The baby sees the lady in the family. The family marries a lady for the triumph. The family quickly wishes the lady vanishes.*

We consider it a non-literary text because it is formed by concatenating simple declarative sentences. This work clearly prioritizes the logical series of events and the existence of a clear narrative arc over text style. We, on the other hand, intend to prioritize a style that shows expressiveness and connects literary images in a new way. Only as a secondary goal do we intend to create a logical sequence of events. This happens to be not just a valid but a common approach in literature, especially from the last century.

Another corpus-based approach that has lately gained prestige because of its good results is the one based on Recurrent Neural Networks (RNN). Especially in the field of NLG, there is interesting work by Karpathy (2015), who trains character language models on RNN's and gets great results at the syntactic level of language. While we consider these results very impressive, we believe that our approach goes beyond the syntax in language. Unlike this kind of work, it intends to find a bridge between syntax and semantics, trying to generate texts which hold both syntactic coherence and meaningful semantic relationships. Karpathy's work manages to generate syntactic style with high fidelity, but he does not give importance to the meaning of the produced text. Conversely, our task is to generate texts that, while maintaining correct syntax, also contain a narrative arc, where a reader could recognize a story.

The present work tries to generate original and novel texts based on a directed combinatorial perspective by exploiting syntactic and semantic patterns found in fiction stories that have 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. Instead, we apply previous knowledge acquired from what we have already read and experienced.

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. Only in the end is a general structure given to the text in order to be able to present it as a story. We were strongly influenced by the NLG architecture presented in (Reiter and Dale, 2000), in that we reuse their concepts, but we try to emphasize text style instead of text planning.

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

2 Literary Structures

We decided to propose a new kind of structures, which will be used as the building blocks of our text's sentences. We did this because canonical syntactic structures are not enough for our purpose of creating novel sentences without entirely losing semantic coherence. 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).

NPV: it is a Noun Phrase that act as an agent in a sentence, together with the action of that agent, the main verb. Examples of NPVs are: "The white-haired woman looked" and "The white-haired woman opened".

VPP: a Verb Phrase holds a verb with its respective complements. Usually, 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 removing the prepositional phrase from the VPs. VPPs only keep verbs with their respective object and a preposition at the end, if it has one. For example, the VP "undertake the development and staffing of the world and its habitants" can be converted into the VPP "undertake the development of".

PPP: a Prepositional Phrase is the circumstance of a VP. For example, if we wish to put a preposition predicate to the VPP "stalked a dozen yards away beneath" the natural question would be "beneath what?" A possible answer could be "beneath the trees".

PHR: Literary texts tend to have subordinate clauses embedded in complex sentences. In order to avoid the complexity of long sentences, and also with the purpose of decontextualizing, we extract simple phrases, subordinates and coordinate phrases into a different literary structure named PHR, in order to be able to combine them also as separate ideas.

CLS: We give the name of special clauses to those clauses that hold an immediate cause-effect relation. Those relations can be found if the sentences containing one or more of the following keywords: so, because, if-then, when, therefore, consequently, hence, for the reason that, as a result, as a consequence.

2.1 Word-Clause Similarity Measure

We will use special clauses (CLS) in the generation step as our story endings. That is why we want to find semantic relatedness between the main word of a story and its possible endings. Hence, a separate database is kept for storing the semantic similarity between words and CLS's. This database contains the most frequent nouns, adjectives and verbs as the keys, and its values are the concatenation of the semantic similarity score, which we call *semsim*, between that word and every CLS that was found. This is obtained with the help of the JCN Similarity Measure (Jiang and Conrath, 1997). With this score, given a word, we can know which are the CLSs semantically closest to that word.

The JCN similarity measure is a semantic metric based on both corpus statistics and lexical taxonomy. It takes advantage of the hierarchical structure that already exists in WordNet (Fellbaum, 1998) and also computes the information content (IC), which is derived from the co-occurrence distribution of that word in a given corpus. The IC of a concept is computed as:

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

In order to avoid polysemy problems, the measure considers not words but particular senses of words (a *synset* in WordNet). The JCN similarity measure can be seen in fact as the inverse of the distance between two synsets, counting the edges linking two senses to their lowest common subsumer (*lcs*) or parent in the WordNet hierarchy and finally incorporating the IC as an important decision factor.

$$JCN_{sim} = \frac{1}{IC(syn_1) + IC(syn_2) - IC(lcs)} \tag{2}$$

We cannot use this measure directly because it can only make comparisons between two senses of words with the same lexical category; so, in order to achieve a comparison between a word and a phrase, we sum the semantic similarity *semsim* (see Equation 4) between the main word W and each of the n words inside the clause whose lexical category matches that of the main word:

$$CLS_{sim} = \sum_{k=0}^{n} semsim(W, w_k) \tag{3}$$

We use the operation *semsim* because JCN measure works with word senses instead of words. 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 each pair of words and keep the highest

score found (that means we keep the highest semantic similarity that two words can have). The operation *semsim* can be defined as:

$$semsim(w_1, w_2) = argmax \left\{ \sum_{i=1}^{x} \sum_{j=1}^{y} jcn_{sim}(s_i, s_j) \right\} \quad (4)$$

where x is the number of senses of w1, and y is the number of senses of w2.

2.2 Sentence Generation

We propose to construct novel sentences based on a simple pattern as shown in Figure 1.

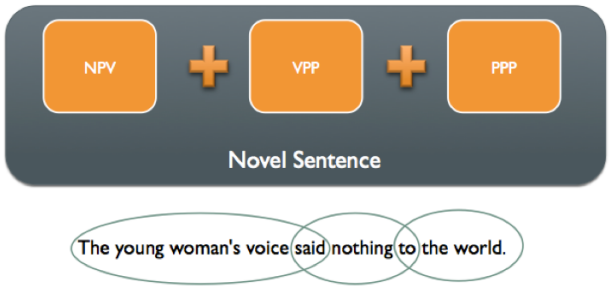

Figure 1: Novel sentences are intersected literary structures.

We can see that the proposed literary structures can be easily concatenated because of their intrinsic properties: An NPV always ends with a verb, a VPP always starts with a verb and ends with a preposition, and a PPP always starts with a preposition, so to construct a sentence, the algorithm just needs to find which of the structures intersect with each other. Given a word, we can have a large number of combinations among the connected phrases. We can start by choosing an NPV, and then look for all the VPPs that start with the same verb as the NPV's 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, with this simple method, we can easily expand phrasal possibilities, starting from a main noun, into several different characters. Then, by looking at VPP intersections we can expand even more from the main verb into several predicates, and finally, if we want a complex predicate, we expand the preposition into many prepositional complements.

2.3 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 the three structures.

Each list of possible structures will be ordered according to a score that our ranking algorithm assigns to them. Every existing structure starts with a score of 1, and points will be added depending on the rules that we describe next. Finally, to retain the creative approach, we run the roulette wheel method in order to choose them. So, the higher-ranked structures will a better chance to be chosen, but each of them has a chance to be chosen anyway.

As an example, we show the set of rules that are followed to score NPV's (Table 1). It is important to mention that each structure follows a different set of rules to get scored, and we defined those rules based on writing manuals such as (Payne, 2011; Espinal et al., 2014; Pinker, 2014).

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 1: Ranking algorithm rules for NPV literary structures.

3 Proposed Architecture

Following the classical NLG process, previous story generation systems just add or remove capabilities to fulfill their specific goals. Besides, they strongly rely on handmade rules, knowledge bases, or schemes 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 ap-

proach, 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 in a bottom-up way. 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.

3.1 Corpus

Our algorithm was trained on a corpus of 9,560 books (which contained both short stories and complete novels) from around 1,100 different authors, which resulted in 1.6GB of text. It contains texts as antique as Homer's "Odyssey" and as present as George R. R. Martin's "Game of Thrones". The most frequent genre was science fiction: We have around 270 texts from Isaac Asimov, 170 from Douglas Adams, 110 from Arthur C. Clarke, for example. There is also classic literature – Honoré de Balzac (78 texts), Mark Twain (110 texts), Charles Dickens (54 texts) – and post-modern literature – 76 texts from William Burroughs and 12 from James Joyce, among many others.

As can be seen above, it is a heterogeneous corpus that covers a wide range of styles from different stages of the history of fiction texts. This allows the algorithm to learn from a wide range of styles, and because of this, our generator can produce texts in such a fashion that it is not easy to typecast it into a certain movement or known style.

The trained corpus resulted in 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). We obtained approximately 7 million structures (3.3 million NPVs, 1.5 million VPPs, 1 million PPPs and PHRs and 60 thousand CLSs).

3.2 Knowledge Base Construction

We created a knowledge base by extracting existing relations of words and style from fiction texts – see Figure 2. We were looking for a mixture of texts that could produce text with a style not directly referable to any previously known story.

We separated sentences that are at most 100 words long, and also 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 in the story generation step. All these deletions, instead of limiting our information, help us to correctly detach even more phrases and isolate them from their original context.

As a first step, before constructing the database, we process the texts using Stanford Parser (Klein and Manning, 2003) in order to work with a set of parsed sentences instead of plain strings. Next, we identify the literary structures. After we have obtained the desired structures from the parsed corpus, we use the Named Entity Recognizer from the NLTK library (Bird et al., 2009) to extract proper names and entities found in text.[2] Then, we apply a ranking algorithm to every extracted structure, so every structure inside our knowledge base will have a score associated with it, in order to improve story coherence and avoid random choice at the generation step. At the same time, we index words and structures in such a fashion that given any word, we can automatically get all its associated structures. Likewise, given any structure index, we can immediately recover the specific list of words that form it. This will help our generation algorithm to navigate the knowledge base fast and also exploit possibilities of avoiding the generation of the same sentences repeatedly on each iteration. Lastly, we implemented our word-clause similarity measure to compute the closest structure for any given word.

We also introduced in the knowledge base a number of lists containing basic grammar details such as words that can receive a proper name, transitive verbs, English prepositions and pronouns. These are

[1] We do use the non-neutral sentences in the Special Clause Extraction, since those are already full sentences, thus they are not building blocks for creating new ones. They are the only sentences inserted in the generated text as they were found in the corpus, meaning that we can trust their syntax.

[2] We realize that the NER from NLTK is trained on news texts; however, we could not find an open library for recognizing entities in fiction texts.

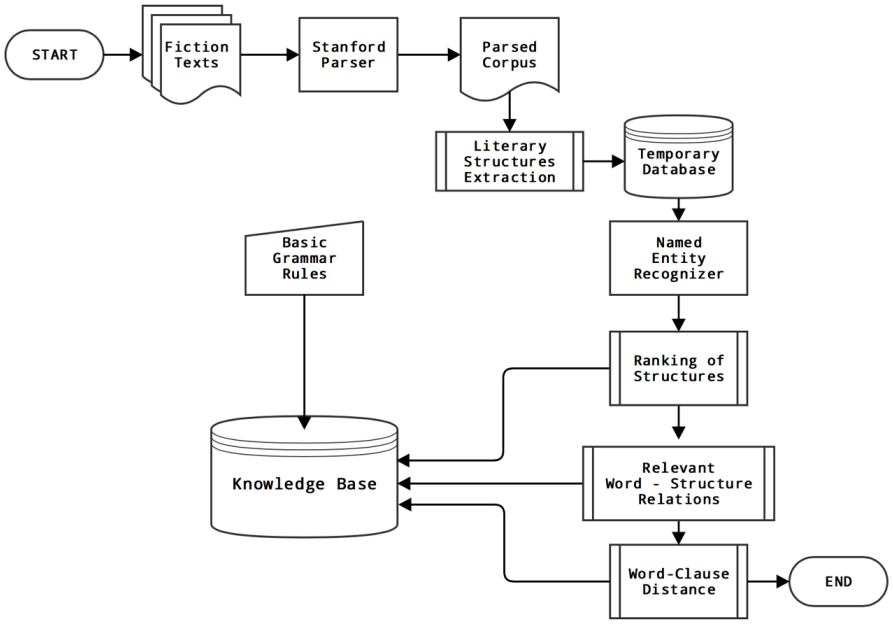

Figure 2: Knowledge Base construction diagram.

used to correct some of the mistakes that may arise while generating our sentences.

3.3 Text Generation Process

Once the knowledge base has been constructed, the first step for generating a story is to provide a word to the algorithm, so it can look for the literary structures that are directly connected to that word in our knowledge base. After extracting the needed structures, the algorithm reads and complements their rankings. Then, it proposes a new way to relate and combine them in order to generate novel sentences that will not only hold a literary style, but also will produce new meanings not necessarily present in 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 directly depend on the corpus, the sequence of sentences that make 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.

We should mention that we merged the modules of Surface Realization and micro-planning, doing both tasks in the same step. Style, content and language production will be managed as a whole, so creativity of 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 macro-planning task to a general schema that every story will follow. This was done 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 micro-planning step, which overshadow the macro structure of the final story.

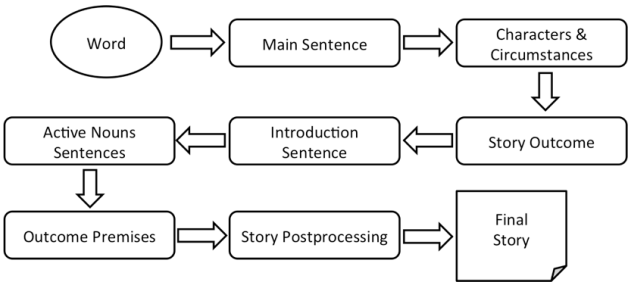

Figure 3: Overview of the text generation step.

4 Building a Story

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. In 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. It 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 (we mainly check verb conjugations, pronoun usage, NER substitutions and determine subjects of sentences for gender and number coherence inside sentences), which we call Story Post-processing. At last, we assemble everything together, and show the final output of the algorithm.

Main Sentence: It is not only the first step, but also the core of every story. Based on the input word, a novel sentence is constructed (NPV+VPP+ PPP). 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.

Characters and Circumstances: In 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 the presentation of the same character with contradictory adjectives (e.g., if we already said that a woman was young, we cannot refer to her as an old woman in later 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 can perform more actions in the future.

Story Outcome: Before elaborating more about our known characters and circumstances, we need to know at what outcome are we aiming at, in order to control a little more the semantics of our sentences. Thus, the next step is to look for a suitable story ending. We rely on our word-clause similarity measure to help us find a meaningful ending to our story. Once the main idea has been created, and based on the vocabulary that is present in it, the algorithm looks for the twenty closest clauses to the main idea. 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 20 top-ranked outcomes (the outcomes closest to our main sentence). Finally, the roulette method is applied to choose among the top-20 list what will become our story outcome.

Introduction Sentence: As we have said before, stories nowadays do not necessarily hold a classical structure of introduction, conflict and resolution. Even so, 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, such as "Once upon a time", "There was once a", etc. This is added as a start in every story, so the reader immediately interprets every sentence as a part of some fiction text.

Active Noun Sentences: The algorithm will use simple phrases (PHRs) in order to go deeper 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 appear 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 with it in our dictionary of characters and circumstances. 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). 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 in order to avoid having a bunch of phrases mentioning diverse nouns and deviating too much from the main idea.

Outcome Premises: To advance the plot even

more, after we described the characters more, we have to look for a connection between the beginning of the story and the outcome. Based on a similar technique as character description, a separate entity dictionary is created for the outcome alone, and the algorithm now looks for the nouns that appear in the selected outcome, and tries to say more about them. 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.

Story Post-processing: The final sentences that will be part of the story have been already generated, but before assembling the story there are a few things to check. They 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 complement, and all prepositions have a nominal phrase as a complement. This is done to avoid having truncated sentences and incomplete ideas. Finally, another important step here is the NER substitution to avoid situations when recognizable entities from other stories are mentioned in ours.

5 Story Example

Now that we have described our methodology to generate stories, we will proceed to show an actual example of the working of the algorithm. We start by generating a completely novel sentence, which will be our main sentence, by the formula NPV+VPP+PPP. The algorithm generates the following:

- NPV: *The first man, yawning, sleepy and bleary-eyed, the lazy beast, stumbled*
- VPP: *stumbled along*
- PPP: *along linguistic pathways that were something other than the most direct ones*

We show next the list of sentences that are concatenated to form a story.

1. **Introduction Sentence**: Once upon a time there was a first man.
2. **Main Sentence**: The first man, yawning, sleepy and bleary-eyed, the lazy beast, stumbled along linguistic pathways that were something other than the most direct ones.
3. **Main Character Name**: His name was Owen.
4. **Most Connected Entity Description**: The great ones had gone to discuss high matters.
5. **Main Word Description**: Owen had brought shame to his door.
6. **Active Noun Descriptions**: The beast was not a showy beast, and it was rather small, having much of the blood.
7. **Outcome Premises**: 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.
8. **Story Outcome**: 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.

We can observe a complete story, made out of structures that were extracted from different sources. Steps 1 and 3 work on templates automatically generated; Step 2 takes three structures (an NPV, a VPP and a PPP), each from a different source; Steps 4 and 5 are two PHRs; Step 6 is a single PHR; Step 7 is made of five concatenated PHRs, each one from a different source text; and finally, Step 8 is one of the 20 CLSs closest to the main word. Our story is then a complete text on its own, an original story, since it was made from 12 different pieces of text that were previously totally unrelated, but now tell a single narrative arc.

6 Experiments and Results

We evaluate the generated stories in a survey where people were asked to score different parameters in texts. 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.

Evaluators: The survey was successfully applied to a population of 32 individuals, all of them students. We considered it important to select individuals who had moderate to strong reading habits. We also kept track of gender (14 women and 18 men) and age balance (people were between 18 and 36 years old). Also the field of study of every evaluator was relevant. Evaluators came from heterogeneous fields such as the 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 in Figure 4.

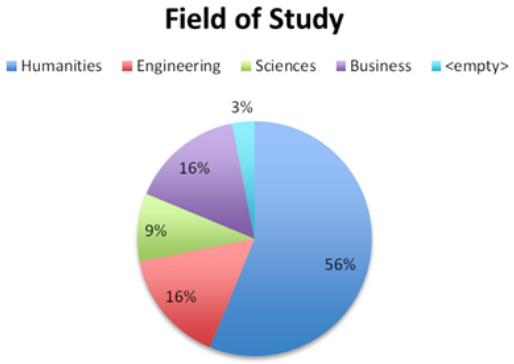

Figure 4: Evaluators' fields of study.

Story Selection: One of the biggest disadvantages of preserving the impression of creativity in the produced texts is this: We do not have entire control of the content and fluency of stories, so the algorithm does not always guarantee to produce a satisfactory text. We countervail this lack of hundred percent guarantee with the ability to produce several stories in short time (approximately five stories per minute). Since it is a new architecture, for now, we settled on producing several stories that could be analyzed by a human reader, who decides which one has achieved the algorithm's result expectations.

For the purpose of evaluation, we generated five instances of stories based on each of the thirty most frequent words (to give the algorithm a broader search space of structures), resulting on a total of 150 stories.[3] Three pre-evaluators read each story and selected those that contained fewer grammatical mistakes, had more lexical variation and a reasonable length. In the end, since we only needed to include four stories in our main survey, the four most voted stories were chosen.

6.1 Evaluation Survey

The survey is inspired by the Turing test, where people have to decide if the text they are reading was produced by another person or by an algorithm. The included texts are shown in Table 2, and judges were not informed about the origin of texts, so they could not know if it was an artificially generated text, or a text by a human author. Additionally, we decided to measure four more parameters: coherence, interest, originality and syntax quality.

Title	Author
Naked Lunch (fragment)	William Burroughs
A Big Man Existed	Machine
The Left Gets Threatened	Rachel Summer
The White Sky Met	Machine
A Beautiful Story	Machine
Finnegans Wake (fragment)	James Joyce
The First Man	Machine

Table 2: Texts that were presented to human judges

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 her opinion is to a given concept. The five questions consist each of a slider between two opposite concepts. The concepts evaluated were coherence, interest, originality and the quality of English. 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.

[3]Please refer to http://ebard.likufanele.com for more examples of generated stories

6.2 Results

We split our survey results in two, that is, we evaluated separately the scores given to human texts and the scores given to artificial texts. As shown in Figure 5, our texts obtained even more positive votes than the human texts. This shows that the evaluators were unable to identify directly any trace of artificiality in our generated texts. On the contrary, they even considered our texts to be more coherent, interesting and with acceptable syntax.

We chose to compare our texts with post-modern authors, because we wanted to emphasize that our texts were at least as experimental as the texts of those authors. This also opens the debate on the characteristics that normally people look for in order to explore outputs generated artificially. The fact that the evaluators scored our texts better than human texts only highlights the fact that people consider an output to be more *human* if they find it more familiar. Since our texts were less elaborate than human texts, evaluators got confused and tried to stick to the more familiar outputs, which happened to be our texts. This makes sense, since our generated texts are only combinations of other ordinary texts. In the end, this successfully proves that our artificial outputs were not recognizable among other literature works.

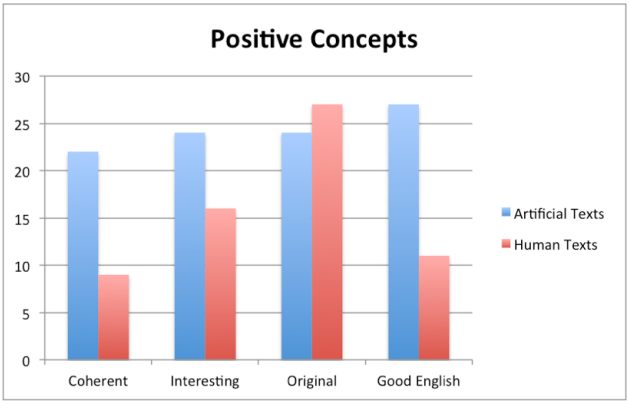

Figure 5: Votes received for positive concepts of each aspect.

To stress even more the creativity (Figure 6) perceived in our texts, we include a more detailed graph showing the percentage of votes that human and artificial texts obtained in each originality interval (we divided the slider in ten intervals to measure this). Recall that the intervals closer to zero mean originality and those closer to ten mean that the texts are common.

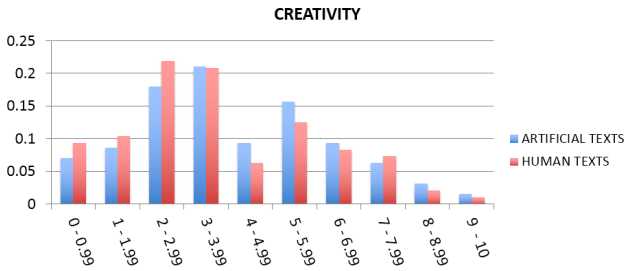

Figure 6: Percentage of votes obtained in the creativity aspect of texts.

There still is one more general chart that shows the overall semantic differential. This is an attempt to measure concept meanings by asking the judge where her position towards two opposite adjectives lie (Osgood et al., 1957). In this case we measured the judges' overall perception of texts. This chart (Figure 7) 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.

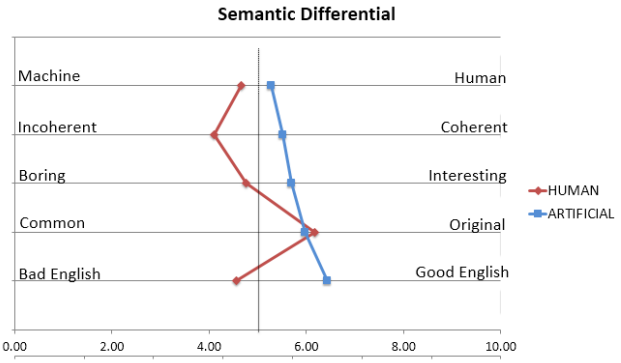

Figure 7: Total semantic differential based on the total population responses.

7 Conclusions

Our evaluation suggests that 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.

The proposed architecture 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 proposed an architecture for generating texts that resemble in a better way the language that is used in fiction texts, and we called this *literary style*. Likewise, we managed to imitate creativity in the sense that we do not have control of the contents of the stories, and it is not possible to know the outcome of the algorithm until it creates a new narrative arc, all of this based on a single word as an input.

Nevertheless, we need to be cautious about our results. We specifically rely on the assumption that most of the meaning that a text holds is created in the mind of the reader. By doing this, we avoid the problem of producing texts with intentionality, and focused on language experiments and guided word combinatory. We successfully avoided the production of text that gives an artificial feeling to the reader, and manages to hold style while being creative at the same time.

References

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Umberto Eco. 2005. Sobre algunas funciones de la literatura. In Helena Lozano Miralles, editor, *Sobre Literatura*, chapter 1, pages 9–23. Debolsillo.

Maria Teresa Espinal, Josep Macia, Jaime Mateu, and Josep Quer. 2014. *Semantica*. Akal.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Pablo Gervás. 2013. Propp's morphology of the folk tale as a grammar for generation. *Computational Models of Narrative*.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings of International Conference Research on Computational Linguistics, 15*.

Andrej Karpathy. 2015. The unreasonable effectiveness of recurrent neural networks. http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430.

Neil McIntyre and Mirella Lapata. 2009. Learning to tell tales: A data-driven approach to story generation. *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics*.

James R. Meehan. 1977. Tale-spin an interactive program that writes stories. *Proceedings of the 5th International Conference on Artificial Intelligence*.

Charles E. Osgood, George Suci, and Percy Tannenbaum. 1957. *The measurement of meaning*. University of Illinois Press.

Thomas E. Payne. 2011. *Understanding English Grammar: a linguistic introduction*. Cambridge University Press.

Rafael Pérez. 1999. Mexica: A computer model of creativity in writing.

Steven Pinker. 2014. *The sense of style: the thinking person's guide to writing in the 21st century*. Penguin Group.

Edhud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Ivo Swartjes and Mariet Theune. 2006. A fabula model for emergent narrative. In *Technologies for Interactive Digital Storytelling and Entertainment Conference*, pages 49–60. Springer.

Perry Thorndyke. 1977. Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, pages 77–110.