
GENERACIÓN AUTOMÁTICA DE FRASES LITERARIAS EN ESPAÑOL

Luis-Gil Moreno-Jiménez

Université d'Avignon/LIA

Universidad Tecnológica de la Selva

luis-gil.moreno-jimenez

@alumni.univ-avignon.fr

Juan-Manuel Torres-Moreno

Université d'Avignon/LIA

Polytechnique Montréal

juan-manuel.torres

@univ-avignon.fr

Roseli S. Wedemann

Universidade do Estado do Rio de Janeiro

roseli@ime.uerj.br

January 31, 2020

ABSTRACT

En este trabajo presentamos un estado del arte en el área de la Creatividad Computacional (CC). En particular abordamos la generación automática de frases literarias en español. Proponemos tres modelos de generación textual basados principalmente en algoritmos estadísticos y análisis sintáctico superficial. Presentamos también algunos resultados preliminares bastante alentadores.

1 Introducción

Los investigadores en Procesamiento de Lenguaje Natural (PLN) durante mucho tiempo han utilizado corpora constituidos por documentos enciclopédicos (notablemente Wikipedia), periodísticos (periódicos o revistas) o especializados (documentos legales, científicos o técnicos) para el desarrollo y pruebas de sus modelos [32, 5, 28].

La utilización y estudios de corpora literarios sistemáticamente han sido dejados a un lado por varias razones. En primer lugar, el nivel de discurso literario es más complejo que los otros géneros. En segundo lugar, a menudo, los documentos literarios hacen referencia a mundos o situaciones imaginarias o alegóricas, a diferencia de los otros géneros que describen sobre todo situaciones o hechos factuales. Estas y otras características presentes en los textos literarios, vuelven sumamente compleja la tarea de análisis automático de este tipo de textos. En este trabajo nos proponemos utilizar corpora literarios, a fin de generar realizaciones literarias (frases nuevas) no presentes en dichos corpora.

La producción de textos literarios es el resultado de un proceso donde una persona hace uso de aptitudes creativas. Este proceso, denominado “proceso creativo”, ha sido analizado por [1], quien propone tres tipos básicos de creatividad: la primera, Creatividad Combinatoria (CCO), donde se fusionan elementos conocidos para la generación de nuevos elementos. La segunda, Creatividad Exploratoria (CE), donde la generación ocurre a partir de la observación o exploración. La tercera, Creatividad Transformacional (CT), donde los elementos generados son producto de alteraciones o experimentaciones aplicadas al dominio de la CE.

Sin embargo, cuando se pretende automatizar el proceso creativo, la tarea debe ser adaptada a métodos formales que puedan ser realizados en un algoritmo. Este proceso automatizado da lugar a un nuevo concepto denominado Creatividad Computacional (CC), introducido por [24], quien retoma para ello la CT y la CE propuestas por [1].

La definición de literatura no tiene un consenso universal, y muchas variantes de la definición pueden ser encontradas. En este trabajo optaremos por introducir una definición pragmática de frase literaria, que servirá para nuestros modelos y experimentos.

Definición. Una frase literaria es una frase que se diferencia de las frases en lengua general, porque contiene elementos (nombres, verbos, adjetivos, adverbios) que son percibidos como elegantes o menos coloquiales que sus equivalentes en lengua general.

En particular, proponemos crear artificialmente frases literarias utilizando modelos generativos y aproximaciones semánticas basados en corpus de lengua literaria. La combinación de esos modelos da lugar a una homosintaxis, es decir, la producción de texto nuevo a partir de formas de discurso de diversos autores. La homosintaxis no tiene el mismo contenido semántico, ni siquiera las mismas palabras, aunque guarda la misma estructura sintáctica.

En este trabajo proponemos estudiar el problema de la generación de texto literario original en forma de frases aisladas, no a nivel de párrafos. La generación de párrafos puede ser objeto de trabajos futuros. Una evaluación de la calidad de las frases generadas por nuestro sistema será presentada.

Este artículo está estructurado como sigue. En la Sección 2 presentamos un estado del arte de la creatividad computacional. En la Sección 3 describimos los corpus utilizados. Nuestros modelos son descritos en la Sección 4. Los resultados y su interpretación se encuentran en la Sección 5. Finalmente la Sección 6 presenta algunas ideas de trabajos futuros antes de concluir.

2 Trabajos previos

La generación de texto es una tarea relativamente clásica, que ha sido estudiada en diversos trabajos. Por ejemplo, [30] presentan un modelo basado en cadenas de Markov para la generación de texto en idioma polaco. Los autores definen un conjunto de estados actuales y calculan la probabilidad de pasar al estado siguiente. La ecuación (1) calcula la probabilidad de pasar al estado X_i a partir de X_j ,

$$P_{ij}(X_i|X_j) = P(X_i \cap X_j)|P(X_j). \quad (1)$$

Para ello, se utiliza una matriz de transición, la cual contiene las probabilidades de transición de un estado actual X_i a los posibles estados futuros X_{i+1} . Cada estado puede estar definido por n -gramas de letras o de palabras.

La tarea inicia en un estado X_i dado por el usuario. Posteriormente, usando la matriz de transición, se calcula la probabilidad de pasar al estado siguiente X_{i+1} . En ese momento el estado predicho X_{i+1} se convierte en el estado actual X_i , repitiendo este proceso hasta satisfacer una condición. Este método tiene un buen comportamiento al generar palabras de 4 o 5 letras. En polaco esta longitud corresponde a la longitud media de la mayor parte de las palabras [31].

También hay trabajos que realizan análisis más profundos para generar no solamente palabras, sino párrafos completos. [29] presentan un algoritmo que genera automáticamente comentarios descriptivos para bloques de código (métodos) en Java. Para ello, se toma el nombre del método y se usa como la acción o idea central de la descripción a generar. Posteriormente se usan un conjunto de heurísticas, para seleccionar las líneas de código del método que puedan aportar mayor información, y se procesan para generar la descripción. La tarea consiste en construir sintagmas, a partir de la idea central dada por el nombre del método, y enriquecerlos con la información de los elementos extraídos. Por ejemplo, si hay un método `removeWall(Wall x)` y se encuentra la llamada al método `removeWall(oldWall)`, la descripción generada podría ser: “Remove old Wall”. Obteniéndose la acción (verbo) y el objeto (sustantivo) directamente del nombre del método y el adjetivo a partir de la llamada. Estas ideas permiten a los autores la generación de comentarios extensos sin perder la coherencia y la gramaticalidad.

También se encuentran trabajos de generación textual que se proponen como meta resultados con un valor más artístico. [25] presentan un conjunto de algoritmos para la generación de una guía narrativa basada en la idea de Creatividad Exploratoria [1]. El modelo establece i/ un conjunto universal U de conceptos relevantes relacionados a un dominio; ii/ un modelo generador de texto; iii/ un subconjunto de conceptos S que pertenecen al conjunto universal U; y iv/ algoritmos encargados de establecer las relaciones entre U y S para generar nuevos conceptos. Estos nuevos conceptos serán posteriormente comparados con los conceptos ya existentes en U para verificar la coherencia y relación con la idea principal. Si los resultados son adecuados, estos nuevos conceptos se utilizan para dar continuación a la narrativa.

Son diversos los trabajos que están orientados a la generación de una narrativa ficticia como cuentos o historias. [4] proponen un modelo de generación de texto narrativo a partir del análisis de *entidades*. Dichas *entidades* son palabras (verbos, sustantivos o adjetivos) dentro de un texto que serán usados para generar la frase siguiente. El modelo recupera las *entidades* obtenidas de tres fuentes principales: la frase actual, la frase previa y el documento completo (contexto), y las procesa con una red neuronal para seleccionar las mejores de acuerdo a diversos criterios. A partir de un conjunto de heurísticas, se analizaron las frases generadas para separar aquellas que expresaran una misma idea (paráfrasis), de aquellas que tuvieran una relación entre sus *entidades* pero con ideas diferentes.

La generación de texto literario es un proceso muy diferente a la generación de texto aleatorio [14, 36] y tampoco se limita a una idea o concepto general. El texto literario está destinado a ser un documento elegante y agradable a la

lectura, haciendo uso de figuras literarias y un vocabulario distinto al empleado en la lengua general. Esto da a la obra una autenticidad y define el estilo del autor. El texto literario también debe diferenciarse de las estructuras rígidas o estereotipadas de los géneros periodístico, enciclopédico o científico.

[37] proponen un modelo para la generación de poemas y se basa en dos premisas básicas: *¿qué decir?* y *¿cómo decirlo?* La propuesta parte de la selección de un conjunto de frases tomando como guía una lista de palabras dadas por el usuario. Las frases son procesadas por un modelo de red neuronal [17], para construir combinaciones coherentes y formular un contexto. Este contexto es analizado para identificar sus principales elementos y generar las líneas del poema, que también pasarán a formar parte del contexto. El modelo fue evaluado manualmente por 30 expertos en una escala de 1 a 5, analizando legibilidad, coherencia y significatividad en frases de 5 palabras, obteniendo una precisión de 0.75. Sin embargo, la coherencia entre frases resultó ser muy pobre.

[10, 11] proponen un modelo de generación de poemas a base de plantillas. El algoritmo inicia con un conjunto de frases relacionadas a partir de palabras clave. Las palabras clave sirven para generar un contexto. Las frases son procesadas usando el sistema PEN¹ para obtener su información gramatical. Esta información es empleada para la generación de nuevas plantillas gramaticales y finalmente la construcción de las líneas del poema, tratando de mantener la coherencia y la gramaticalidad.

El modelo sentiGAN [33] pretende generar texto con un contexto emocional. Se trata de una actualización del modelo GAN (*Generative Adversarial Net*) [12] que ha producido resultados alentadores en la generación textual, aunque con ciertos problemas de calidad y coherencia. Se utiliza el análisis semántico de una entrada proporcionada por el usuario que sirve para la creación del contexto. La propuesta principal de SentiGAN sugiere establecer un número definido de generadores textuales que deberán producir texto relacionado a una emoción definida. Los generadores son entrenados bajo dos esquemas: i/ una serie de elementos lingüísticos que deben ser evitados para la generación del texto; y ii/ un conjunto de elementos relacionados con la emoción ligada al generador. A través de cálculos de distancia, heurísticas y modelos probabilísticos, el generador crea un texto lo más alejado del primer esquema y lo más cercano al segundo.

También existen trabajos con un alcance más corto pero de mayor precisión. [13] proponen la evaluación de un conjunto de datos con un modelo basado en redes neuronales para la generación de subconjuntos de multi-palabras. Este mismo análisis, se considera en [9], en donde se busca establecer o detectar la relación hiperónimo-hipónimo con la ayuda del modelo de *Deep Learning* Word2vec [18]. La propuesta de [9] reporta una precisión de 0.70 al ser evaluado sobre un corpus manualmente etiquetado.

La literatura es una actividad artística que exige capacidades creativas importantes y que ha llamado la atención de científicos desde hace cierto tiempo. [24] realiza un estado del arte interesante donde menciona algunos trabajos que tuvieron un primer acercamiento a la obra literaria desde una perspectiva superficial. Por ejemplo, el modelo “Through the park” [20], es capaz de generar narraciones históricas empleando la elipsis. Esta técnica es empleada para manipular, entre otras cosas, el ritmo de la narración. En los trabajos “About So Many Things” [21] y “Taroko Gorge” [22] se muestran textos generados automáticamente. El primero de ellos genera estrofas de 4 líneas estrechamente relacionadas entre ellas. Eso se logra a través de un análisis gramatical que establece conexiones entre entidades de distintas líneas. El segundo trabajo muestra algunos poemas cortos generados automáticamente con una estructura más compleja que la de las estrofas. El inconveniente de ambos enfoques es el uso de una estructura inflexible, lo que genera textos repetitivos con una gramaticalidad limitada.

El proyecto MEXICA modela la generación colaborativa de narraciones [24]. El propósito es la generación de narraciones completas utilizando obras de la época Precolombina. MEXICA genera narraciones simulando el proceso creativo de E-R (*Engaged* y *Reflexive*) [26]. Este proceso se describe como la acción, donde el autor trae a su mente un conjunto de ideas y contextos y establece una conexión coherente entre estas (E). Posteriormente se reflexiona sobre las conexiones establecidas y se evalúa el resultado final para considerar si este realmente satisface lo esperado (R). El proceso itera hasta que el autor lo considera concluido.

3 Corpus utilizados

3.1 Corpus 5KL

Este corpus fue constituido con aproximadamente 5 000 documentos (en su mayor parte libros) en español. Los documentos originales, en formatos heterogéneos, fueron procesados para crear un único documento codificado en *utf8*. Las frases fueron segmentadas automáticamente, usando un programa en PERL 5.0 y expresiones regulares, para obtener una frase por línea.

¹Disponible en: <http://code.google.com/p/pen>

Las características del corpus 5KL se encuentran en la Tabla 1. Este corpus es empleado para el entrenamiento de los modelos de aprendizaje profundo (*Deep Learning*, Sección 4).

	Frases	Palabras	Caracteres
5KL	9 M	149 M	893 M
Media por documento	2.4 K	37.3 K	223 K

Table 1: Corpus 5KL compuesto de 4 839 obras literarias.

El corpus literario 5KL posee la ventaja de ser muy extenso y adecuado para el aprendizaje automático. Tiene sin embargo, la desventaja de que no todas las frases son *necesariamente* “frases literarias”. Muchas de ellas son frases de lengua general: estas frases a menudo otorgan una fluidez a la lectura y proporcionan los enlaces necesarios a las ideas expresadas en las frases literarias.

Otra desventaja de este corpus es el ruido que contiene. El proceso de segmentación puede producir errores en la detección de fronteras de frases. También los números de página, capítulos, secciones o índices producen errores. No se realizó ningún proceso manual de verificación, por lo que a veces se introducen informaciones indeseables: *copyrights*, datos de la edición u otros. Estas son, sin embargo, las condiciones que presenta un corpus literario real.

3.2 Corpus 8KF

Un corpus heterogéneo de casi 8 000 frases literarias fue constituido manualmente a partir de poemas, discursos, citas, cuentos y otras obras. Se evitaron cuidadosamente las frases de lengua general, y también aquellas demasiado cortas ($N \leq 3$ palabras) o demasiado largas ($N \geq 30$ palabras). El vocabulario empleado es complejo y estético, además que el uso de ciertas figuras literarias como la rima, la anáfora, la metáfora y otras pueden ser observadas en estas frases.

Las características del corpus 8KF se muestran en la Tabla 2. Este corpus fue utilizado principalmente en los dos modelos generativos: modelo basado en cadenas de Markov (Sección 4.1) y modelo basado en la generación de Texto enlatado (*Canned Text*, Sección 4.2).

	Frases	Palabras	Caracteres
8KF	7 679	114 K	652 K
Media por frase	–	15	85

Table 2: Corpus 8KF compuesto de 7 679 frases literarias.

4 Modelos propuestos

En este trabajo proponemos tres modelos híbridos (combinaciones de modelos generativos clásicos y aproximaciones semánticas) para la producción de frases literarias. Hemos adaptado dos modelos generativos, usando análisis sintáctico superficial (*shallow parsing*) y un modelo de aprendizaje profundo (*Deep Learning*) [7], combinados con tres modelos desarrollados de aproximación semántica.

En una primera fase, los modelos generativos recuperan la información gramatical de cada palabra del corpus 8KF (ver Sección 3), en forma de etiquetas POS (*Part of Speech*), a través de un análisis morfosintáctico. Utilizamos Freeling [23] que permite análisis lingüísticos en varios idiomas². Por ejemplo, para la palabra “Profesor” Freeling genera la etiqueta POS [NCMS000]. La primera letra indica un sustantivo (Noun), la segunda un sustantivo común (Common); la tercera indica el género masculino (Male) y la cuarta da información de número (Singular). Los 3 últimos caracteres dan información detallada del campo semántico, entidades nombradas, etc.³ En nuestro caso usaremos solamente los 4 primeros niveles de las etiquetas.

Con los resultados del análisis morfosintáctico, se genera una salida que llamaremos *Estructura gramatical vacía* (EGV): compuesta exclusivamente de una secuencia de etiquetas POS; o *Estructura gramatical parcialmente vacía* (EGP), compuesta de etiquetas POS y de palabras funcionales (artículos, pronombres, conjunciones, etc.).

²Puede ser obtenido en la dirección: <http://nlp.lsi.upc.edu/freeling>

³Más detalles de las etiquetas Freeling en <http://blade10.cs.upc.edu/freeling-old/doc/tagsets/tagset-es.html>

En la segunda fase, las etiquetas POS (en la EGV y la EGP) serán reemplazadas por un vocabulario adecuado usando ciertas aproximaciones semánticas.

La producción de una frase $f(Q, N)$ es guiada por dos parámetros: un contexto representado por un término Q (o *query*) y una longitud $3 \leq N \leq 15$, dados por el usuario. Los corpus 5KL y 8KF son utilizados en varias fases de la producción de las frases f .

- El Modelo 1 está compuesto por: i/ un modelo generativo estocástico basado en cadenas de Markov para la selección de la próxima etiqueta POS usando el algoritmo de Viterbi; y ii/ un modelo de aprendizaje profundo (Word2vec), para recuperar el vocabulario que reemplazará la secuencia de etiquetas POS.
- El Modelo 2 es una combinación de: i/ el modelo generativo de Texto enlatado; y ii/ un modelo Word2vec, con un cálculo de distancias entre diversos vocabularios que han sido constituidos manualmente.
- El Modelo 3 utiliza: i/ la generación de Texto enlatado; y ii/ una interpretación geométrica del aprendizaje profundo. Esta interpretación está basada en una búsqueda de información iterativa (*Information Retrieval*, IR), que realiza simultáneamente un alejamiento de la semántica original y un acercamiento al *query* Q del usuario.

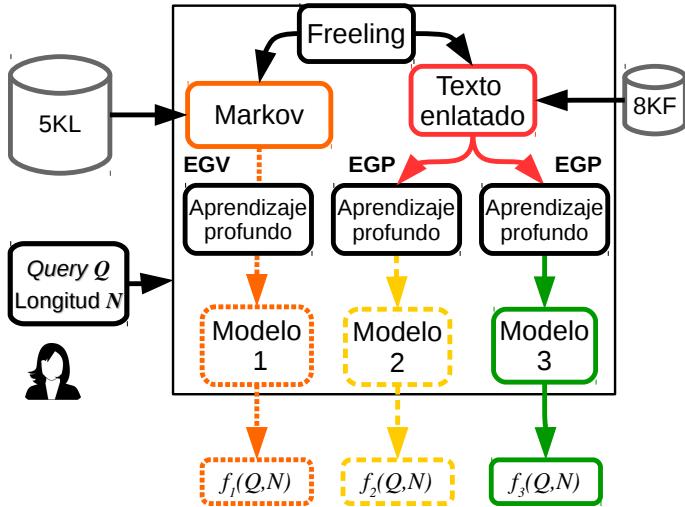


Figure 1: Arquitectura general de los modelos.

4.1 Modelo generativo estocástico usando cadenas de Markov

Este modelo generativo, que llamaremos *Modelo de Markov*, está basado en el algoritmo de Viterbi y las cadenas de Markov [15], donde se selecciona una etiqueta POS con la máxima probabilidad de ocurrencia, para ser agregada al final de la secuencia actual.

Utilizamos el corpus de frases literarias 8KF (ver Sección 3.2), que fue convenientemente filtrado para eliminar *tokens* indeseables: números, siglas, horas y fechas. El corpus filtrado se analizó usando Freeling, que recibe en entrada una cadena de texto y entrega el texto con una etiqueta POS para cada palabra. El corpus es analizado frase a frase, reemplazando cada palabra por su respectiva etiqueta POS. Al final del análisis, se obtiene un nuevo corpus 8KPOS con $s = 7\,679$ secuencias de etiquetas POS, correspondientes al mismo número de frases del corpus 8KF. Las secuencias del corpus 8KPOS sirven como conjunto de entrenamiento para el algoritmo de Viterbi, que calcula las probabilidades de transición, que serán usadas para generar cadenas de Markov.

Las s estructuras del corpus 8KPOS procesadas con el algoritmo de Viterbi son representadas en una matriz de transición $P_{[s \times s]}$. P será utilizada para crear nuevas secuencias de etiquetas POS no existentes en el corpus 8KPOS, simulando un proceso creativo. Nosotros hemos propuesto el algoritmo *Creativo-Markov* que describe este procedimiento.

En este algoritmo, X_i representa el estado de una etapa de la creación de una frase, en el instante i , que corresponde a una secuencia de etiquetas POS. Siguiendo un procedimiento de Markov, en un instante i se selecciona la próxima etiqueta POS_{i+1} , con máxima probabilidad de ocurrencia, dada la última etiqueta POS_i de la secuencia X_i . La etiqueta POS_{i+1} será agregada al final de X_i para generar el estado X_{i+1} . $P(X_{i+1} = Y | X_i = Z)$ es la probabilidad de

transición de un estado a otro, obtenido con el algoritmo de Viterbi. Se repiten las transiciones, hasta alcanzar una longitud deseada.

El resultado es una EGV, donde cada cuadro vacío representa una etiqueta POS que será remplazada por una palabra en la etapa final de generación de la nueva frase. El remplazo se realiza usando un modelo de aprendizaje profundo (Sección 4.3). La arquitectura general de este modelo se muestra en la Figura 2.

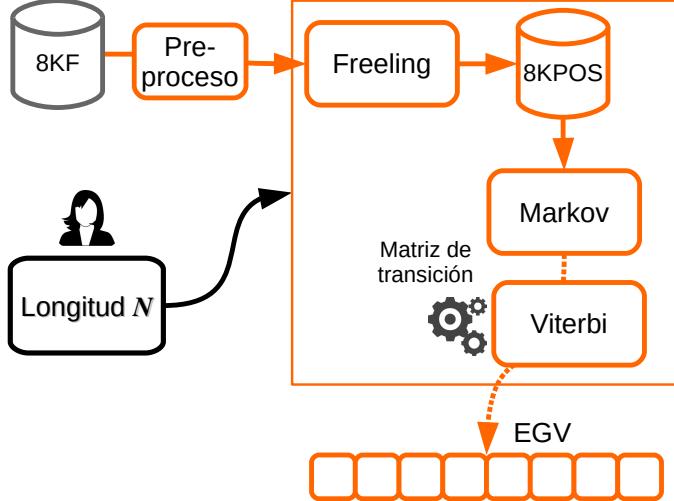


Figure 2: Modelo generativo estocástico (Markov) que produce una estructura gramatical vacía EGV.

4.2 Modelo generativo basado en Texto enlatado

El algoritmo *creativo-Markov* del *Modelo de Markov* logra reproducir patrones lingüísticos (secuencias POS) detectados en el corpus 8KPOS, pero de corta longitud. Cuando se intentó extender la longitud de las frases a $N > 6$ palabras, no fue posible mantener la coherencia y legibilidad (como se verá en la Sección 4.3). Decidimos entonces utilizar métodos de generación textual guiados por estructuras morfosintácticas fijas: el Texto enlatado. [19] argumentan que el uso de estas estructuras ahorran tiempo de análisis sintáctico y permite concentrarse directamente en el vocabulario.

La técnica de Texto enlatado ha sido empleada también en varios trabajos, con objetivos específicos. [16, 6] desarrollaron modelos para la generación de diálogos y frases simples. Esta técnica es llamada “Generación basada en plantillas” (*Template-based Generation*) o de manera intuitiva, Texto enlatado⁴.

Decidimos emplear texto enlatado para la generación textual usando un corpus de plantillas (*templates*), construido a partir del corpus 8KF (Sección 3). Este corpus contiene estructuras gramaticales flexibles que pueden ser manipuladas para crear nuevas frases. Estas plantillas pueden ser seleccionadas aleatoriamente o a través de heurísticas, según un objetivo predefinido.

Una plantilla es construida a partir de las palabras de una frase f , donde se reemplazan únicamente las palabras llenas de las clases verbo, sustantivo o adjetivo $\{V, S, A\}$, por sus respectivas etiquetas POS. Las otras palabras, en particular las palabras funcionales, son conservadas. Esto producirá una *estructura gramatical parcialmente vacía*, *EGP*. Posteriormente las etiquetas podrán ser reemplazadas por palabras (términos), relacionadas con el contexto definido por el *query Q* del usuario.

El proceso inicia con la selección aleatoria de una frase original $f_o \in$ corpus 8KF de longitud $|f_o| = N$. f_o será analizada con Freeling para identificar los sintagmas. Los elementos $\{V, S, A\}$ de los sintagmas de f_o serán reemplazados por sus respectivas etiquetas POS. Estos elementos son los que mayor información aportan en cualquier texto, independientemente de su longitud o género [2]. Nuestra hipótesis es que al cambiar solamente estos elementos, simulamos la generación de frases por homosintaxis: semántica diferente, misma estructura⁵.

La salida de este proceso es una estructura híbrida parcialmente vacía (EGP) con palabras funcionales que dan un soporte gramatical y las etiquetas POS. La arquitectura general de este modelo se ilustra en la Figura 3. Los cuadros llenos representan palabras funcionales y los cuadros vacíos etiquetas POS a ser reemplazadas.

⁴<http://projects.ict.usc.edu/nld/cs599s13/LectureNotes/cs599s13dialogue2-13-13.pdf>

⁵Al contrario de la paráfrasis que busca conservar completamente la semántica, alterando completamente la estructura sintáctica.

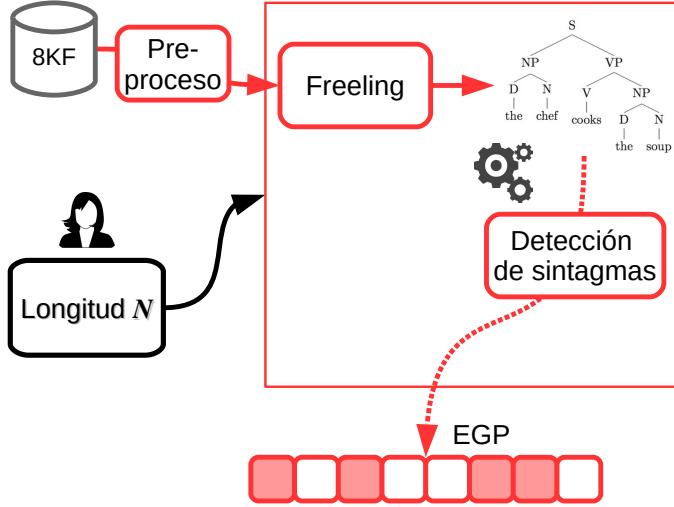


Figure 3: Modelo generativo de Texto enlatado que produce una estructura parcialmente vacía.

4.3 Modelo 1: Markov y aprendizaje profundo

Los modelos generativos generan estructuras gramaticales vacías (EGV) o parcialmente vacías (EGP) que pueden ser manipuladas para generar nuevas frases $f(Q, N)$. La idea es que las frases f sean generadas por homosintaxis. En esta sección, proponemos un modelo de aproximación semántica que utiliza el algoritmo Word2vec (basado en aprendizaje profundo), combinado con el modelo generativo de Markov descrito en la Sección 4.1. El proceso se describe a continuación.

El corpus 5KL es pre-procesado para uniformizar el formato del texto, eliminando caracteres que no son importantes para el análisis semántico: puntuación, números, etc. Esta etapa prepara los datos de entrenamiento del algoritmo de aprendizaje profundo que utiliza una representación vectorial del corpus 5KL. Para el aprendizaje profundo utilizamos la biblioteca Gensim⁶, la versión en Python de Word2vec⁷. Con este algoritmo se obtiene un conjunto de palabras asociadas (*embeddings*) a un contexto definido por un *query* Q . Word2vec recibe un término Q y devuelve un léxico $L(Q) = (w_1, w_2, \dots, w_m)$ que representa un conjunto de m palabras semánticamente próximas a Q . Formalmente, Word2vec: $Q \rightarrow L(Q)$.

El próximo paso consiste en procesar la EGV producida por Markov. Las etiquetas POS serán identificadas y clasificadas como POS_Φ funcionales (correspondientes a puntuación y palabras funcionales) y POS_λ llenas $\in \{V, S, A\}$ (verbos, sustantivos, adjetivos).

Las etiquetas POS_Φ serán reemplazadas por palabras obtenidas de recursos lingüísticos (diccionarios) construidos con la ayuda de Freeling. Los diccionarios consisten en entradas de pares: POS_Φ y una lista de palabras y signos asociados, formalmente $\text{POS}_\Phi \rightarrow l(\text{POS}_\Phi) = (l_1, l_2, \dots, l_j)$. Se reemplaza aleatoriamente cada POS_Φ por una palabra de l que corresponda a la misma clase gramatical.

Las etiquetas POS_λ serán reemplazadas por las palabras producidas por Word2vec $L(Q)$. Si ninguna de las palabras de $L(Q)$ tiene la forma sintáctica exigida por POS_λ , empleamos la biblioteca PATTERN⁸ para realizar conjugaciones o conversiones de género y/o número y reemplazar correctamente POS_λ .

Si el conjunto de palabras $L(Q)$, no contiene ningún tipo de palabra llena, que sea adecuada o que pueda manipularse con la biblioteca PATTERN, para reemplazar las etiquetas POS_λ , se toma otra palabra, $w_i \in L(Q)$, lo más cercana a Q (en función de la distancia producida por Word2vec). Se define un nuevo $Q^* = w_i$ que será utilizado para generar un nuevo conjunto de palabras $L(Q^*)$. Este procedimiento se repite hasta que $L(Q^*)$ contenga una palabra que pueda reemplazar la POS_λ en cuestión. El resultado de este procedimiento es una nueva frase f que no existe en los corpora 5KL y 8KF. La Figura 4 muestra el proceso descrito.

⁶Disponible en: <https://pypi.org/project/gensim/>

⁷<https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa>

⁸<https://www.clips.uantwerpen.be/pattern>

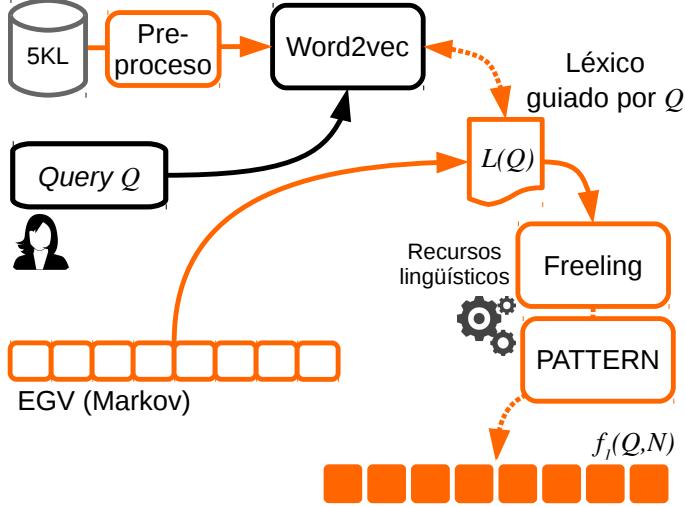


Figure 4: Modelo 1: Aproximación semántica usando Markov y aprendizaje profundo.

4.4 Modelo 2: Texto enlatado, aprendizaje profundo y análisis morfosintáctico

En este modelo proponemos una combinación entre el modelo de Texto enlatado (Sección 4.2) y un algoritmo de aprendizaje profundo con Word2vec entrenado sobre el corpus 5KL. El objetivo es eliminar las iteraciones del Modelo 1, que son necesarias cuando las etiquetas POS⁹ no pueden ser reemplazadas con el léxico $L(Q)$.

Se efectúa un análisis morfosintáctico del corpus 5KL usando Freeling y se usan las etiquetas POS para crear conjuntos de palabras que posean la misma información gramatical (etiquetas POS idénticas). Una Tabla Asociativa (TA) es generada como resultado de este proceso. La TA consiste en k entradas de pares POS_k y una lista de palabras asociadas. Formalmente $POS_k \rightarrow V_k = \{v_{k,1}, v_{k,2}, \dots, v_{k,i}\}$.

El Modelo 2 es ejecutado una sola vez para cada etiqueta POS_k . La EGP no será reemplazada completamente: las palabras funcionales y los signos de puntuación son conservados.

Para generar una nueva frase se reemplaza cada etiqueta $POS_k \in EGP$, $k = 1, 2, \dots$, por una palabra adecuada. Para cada etiqueta POS_k , se recupera el léxico V_k a partir de TA.

El vocabulario es procesado por el algoritmo Word2vec, que calcula el valor de proximidad (distancia) entre cada palabra del vocabulario $v_{k,i}$ y el query Q del usuario, $dist(Q, v_{k,i})$. Después se ordena el vocabulario V_k en forma descendente según los valores de proximidad $dist(Q, v_{k,i})$ y se escoge aleatoriamente uno de los primeros tres elementos para reemplazar la etiqueta POS_k de la EGP.

El resultado es una nueva frase $f_2(Q, N)$ que no existe en los corpora 5KL y 8KF. El proceso se ilustra en la figura 5.

4.5 Modelo 3: Texto enlatado, aprendizaje profundo e interpretación geométrica

El Modelo 3 reutiliza varios de los recursos anteriores: el algoritmo Word2vec, la Tabla Asociativa TA y la estructura grammatical parcialmente vacía (EGP) obtenida del modelo de Texto enlatado. El modelo utiliza distancias vectoriales para determinar las palabras más adecuadas que sustituirán las etiquetas POS de una EGP y así generar una nueva frase. Para cada etiqueta POS_k , $k = 1, 2, \dots \in EGP$, que se desea sustituir, usamos el algoritmo descrito a continuación.

Se construye un vector para cada una de las tres palabras siguientes:

- o : es la palabra k de la frase f_o (Sección 4.2), correspondiente a la etiqueta POS_k . Esta palabra permite recrear un contexto del cual la nueva frase debe alejarse, evitando producir una paráfrasis.
- Q : palabra que define al query proporcionado por el usuario.
- w : palabra candidata que podría reemplazar POS_k , $w \in V_k$. El vocabulario posee un tamaño $|V_k| = m$ palabras y es recuperado de la TA correspondiente a la POS_k .

⁹Por motivos de claridad de la notación, en esta sección y en la siguiente una etiqueta POS_λ será designada solamente por POS.

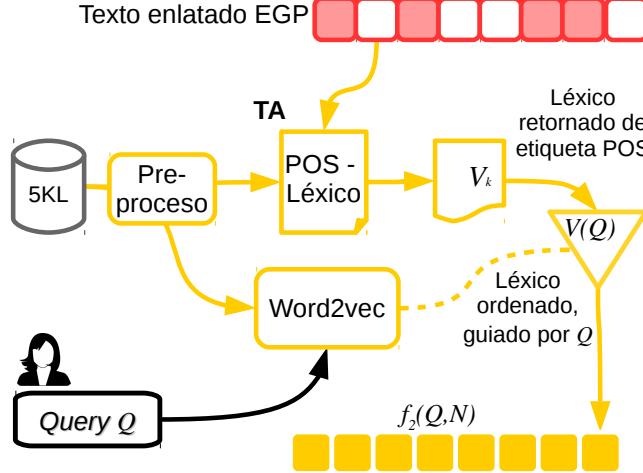


Figure 5: Modelo 2: Aproximación semántica basada en *Deep Learning* y análisis morfosintáctico.

Las 10 palabras o_i más próximas a o , las 10 palabras Q_i más próximas a Q y las 10 palabras w_i más próximas a w (en este orden y obtenidas con Word2vec), son concatenadas y representadas en un vector simbólico \vec{U} de 30 dimensiones. El número de dimensiones fue fijado a 30 de manera empírica, como un compromiso razonable entre diversidad léxica y tiempo de procesamiento. El vector \vec{U} puede ser escrito como:

$$\vec{U} = [u_1, \dots, u_{10}, u_{11}, \dots, u_{20}, u_{21}, \dots, u_{30}], \quad (2)$$

donde cada elemento $u_j, j = 1, \dots, 10$, representa una palabra próxima a o ; $u_j, j = 11, \dots, 20$, representa una palabra próxima a Q ; y $u_j, j = 21, \dots, 30$, es una palabra próxima a w . \vec{U} puede ser re-escrito de la siguiente manera (ecuación 3):

$$\vec{U} = [o_1, \dots, o_{10}, Q_{11}, \dots, Q_{20}, w_{21}, \dots, w_{30}]. \quad (3)$$

o, Q y w generan respectivamente tres vectores numéricos de 30 dimensiones:

$$\begin{aligned} o : \vec{X} &= [x_1, \dots, x_{10}, x_{11}, \dots, x_{20}, x_{21}, \dots, x_{30}] \\ Q : \vec{Q} &= [q_1, \dots, q_{10}, q_{11}, \dots, q_{20}, q_{21}, \dots, q_{30}] \\ w : \vec{W} &= [w_1, \dots, w_{10}, w_{11}, \dots, w_{20}, w_{21}, \dots, w_{30}] \end{aligned}$$

donde los valores de \vec{X} son obtenidos tomando la distancia entre la palabra o y cada palabra $u_j \in \vec{U}, j = 1, \dots, 30$. La distancia, $x_j = dist(o, u_j)$ es proporcionada por Word2vec y además $x_j \in [0, 1]$. Evidentemente la palabra o estará más próxima a las 10 primeras palabras u_j que a las restantes.

Un proceso similar permite obtener los valores de \vec{Q} y \vec{W} a partir de Q y w , respectivamente. En estos casos, el *query* Q estará más próximo a las palabras u_j en las posiciones $j = 11, \dots, 20$ y la palabra candidata w estará más próxima a las palabras u_j en las posiciones $j = 21, \dots, 30$.

Enseguida, se calculan las similitudes coseno entre \vec{Q} y \vec{W} (ecuación 4) y entre \vec{X} y \vec{W} (ecuación 5). Estos valores también están normalizados entre [0,1].

$$\theta = \cos(\vec{Q}, \vec{W}) = \frac{\vec{Q} \cdot \vec{W}}{|\vec{Q}| |\vec{W}|} \quad (4)$$

$$\beta = \cos(\vec{X}, \vec{W}) = \frac{\vec{X} \cdot \vec{W}}{|\vec{X}| |\vec{W}|} \quad (5)$$

El proceso se repite para todas las palabras w del léxico V_k . Esto genera otro conjunto de vectores \vec{X}, \vec{Q} y \vec{W} para los cuales se deberán calcular nuevamente las similitudes. Al final se obtienen m valores de similitudes θ_i y $\beta_i, i = 1, \dots, m$, y se calculan los promedios $\langle \theta \rangle$ y $\langle \beta \rangle$.

El cociente normalizado $\left(\frac{\langle \theta \rangle}{\theta_i} \right)$ indica qué tan grande es la similitud de θ_i con respecto al promedio $\langle \theta \rangle$ (interpretación de tipo maximización); es decir, que tan próxima se encuentra la palabra candidata w al *query* Q .

El cociente normalizado $\left(\frac{\beta_i}{\langle \beta \rangle}\right)$ indica qué tan reducida es la similitud de β_i con respecto a $\langle \beta \rangle$ (interpretación de tipo minimización); es decir, qué tan lejos se encuentra la palabra candidata w de la palabra o de f_o .

Estas fracciones se obtienen en cada par (θ_i, β_i) y se combinan (minimización-maximización) para calcular un score S_i , según la ecuación (6):

$$S_i = \left(\frac{\langle \theta \rangle}{\theta_i} \right) \cdot \left(\frac{\beta_i}{\langle \beta \rangle} \right) \quad (6)$$

Mientras más elevado sea el valor S_i , mejor obedece a nuestros objetivos: acercarse al *query* y alejarse de la semántica original.

Finalmente ordenamos en forma decreciente la lista de valores de S_i y se escoge, de manera aleatoria, entre los 3 primeros, la palabra candidata w que reemplazará la etiqueta POS_k en cuestión. El resultado es una nueva frase $f_3(Q, N)$ que no existe en los corpora utilizados para construir el modelo.

En la Figura 6 se muestra una representación del modelo descrito.

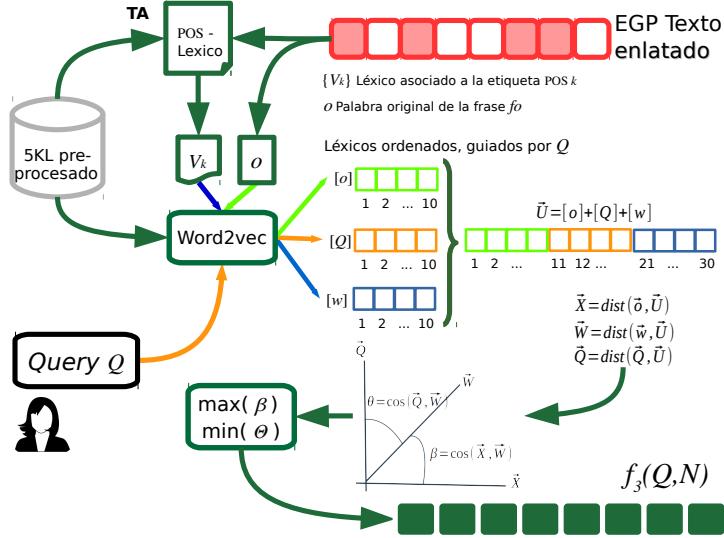


Figure 6: Modelo 3: Aproximación semántica basada en interpretación geométrica min-max.

5 Experimentos y resultados

Dado la especificidad de nuestros experimentos (idioma, corpora disponibles, homosintaxis), no es posible compararse directamente con otros métodos.

Tampoco consideramos la utilización de un *baseline* de tipo aleatorio, porque los resultados carecerían de la homosintaxis y sería sumamente fácil obtener mejores resultados. Dicho lo anterior, el Modelo 1 podría ser considerado como nuestro propio *baseline*.

5.1 Resultados

A continuación presentamos un protocolo de evaluación manual de los resultados obtenidos. El experimento consistió en la generación de 15 frases por cada uno de los tres modelos propuestos. Para cada modelo, se consideraron tres *queries*: $Q = \{\text{AMOR, GUERRA, SOL}\}$, generando 5 frases con cada uno. Las 15 frases fueron mezcladas entre sí y reagrupadas por *queries*, antes de presentarlas a los evaluadores.

Para la evaluación, se pidió a 7 personas leer cuidadosamente las 45 frases (15 frases por *query*). Todos los evaluadores poseen estudios universitarios y son hispanohablantes nativos. Se les pidió anotar en una escala de [0,1,2] (donde 0=mal, 1=aceptable y 2=correcto) los criterios siguientes:

- **Gramaticalidad:** ortografía, conjugaciones correctas, concordancia en género y número.
- **Cohesión:** legibilidad, percepción de una idea general.

- **Contexto:** relación de la frase con respecto al *query*.

Los resultados de la evaluación se presentan en la Tabla 3, en la forma de promedios normalizados entre [0,1] y de su desviación estándar σ .

Modelo	1	2	3
Gramaticalidad	0.55	0.74	0.77
σ	± 0.18	± 0.11	± 0.13
Coherencia	0.25	0.56	0.60
σ	± 0.15	± 0.11	± 0.14
Contexto	0.67	0.35	0.53
σ	± 0.25	± 0.17	± 0.19

Table 3: Resultados de la evaluación manual.

Las frases generadas por los modelos propuestos presentan características particulares.

El Modelo 1 produce generalmente frases con un contexto estrechamente relacionado con el *query* del usuario, pero a menudo carecen de coherencia y gramaticalidad. Este modelo presenta el valor más alto para el contexto, pero también la desviación estándar más elevada. Se puede inferir que existe cierta discrepancia entre los evaluadores. Los valores altos para el contexto se explican por el grado de libertad de la EGV generada por el modelo de Markov. La EGV permite que todos los elementos de la estructura puedan ser sustituidos por un léxico guiado únicamente por los resultados del algoritmo Word2vec.

El Modelo 2 genera frases razonablemente coherentes y grammaticalmente correctas, pero en ocasiones el contexto se encuentra más próximo a la frase original que al *query*. Esto puede ser interpretado como una paráfrasis elemental, que no es lo que deseamos.

Finalmente, el Modelo 3 genera frases coherentes, grammaticalmente correctas y mejor relacionadas al *query* que el Modelo 2. Esto se logra siguiendo una intuición opuesta a la paráfrasis: buscamos conservar la estructura sintáctica de la frase original, generando una semántica completamente diferente.

Por otro lado, la mínima dispersión se observa en el Modelo 1, es decir, hay una gran concordancia entre las percepciones de los evaluadores para este criterio.

6 Conclusión y trabajo futuro

En este artículo hemos presentado tres modelos de producción de frases literarias. La generación de este género textual necesita sistemas específicos que deben considerar el estilo, la sintaxis y una semántica que no necesariamente respeta la lógica de los documentos de géneros factuales, como el periodístico, enciclopédico o científico. Los resultados obtenidos son alentadores para el Modelo 3, utilizando Texto enlatado, aprendizaje profundo y una interpretación del tipo IR. El trabajo a futuro necesita la implementación de módulos para procesar los *queries* multi-término del usuario. También se tiene contemplada la generación de frases retóricas utilizando los modelos aquí propuestos u otros con un enfoque probabilístico [3]. Los modelos aquí presentados pueden ser enriquecidos a través de la integración de otros componentes, como características de una personalidad y/o las emociones [34, 35, 8, 27]. Finalmente, un protocolo de evaluación semi-automático (y a gran escala) está igualmente previsto.

Agradecimientos

Los autores agradecen a Eric SanJuan respecto a las ideas y el concepto de la homosintaxis.

References

- [1] Margaret A Boden. *The creative mind: Myths and Mechanisms*. Routledge, 2004.
- [2] David B Bracewell, Fuji Ren, and Shingo Kuriowa. Multilingual single document keyword extraction for information retrieval. In *2005 International Conference on Natural Language Processing and Knowledge Engineering*, pages 517–522. IEEE, 2005.

- [3] Eric Charton and Juan-Manuel Torres-Moreno. Modélisation automatique de connecteurs logiques par analyse statistique du contexte. *Revue Canadienne de Sciences et de Bibliothéconomie (RCSIB) / Canadian Journal of Information and Library Science*, 35:287–306, 01 2010.
- [4] Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. Neural text generation in stories using entity representations as context. In *NAACL: Human Language Technologies*, vol. 1, pages 2250–2260, New Orleans, Louisiana, June 2018. ACL.
- [5] Iria da Cunha, M. Teresa Cabré, Eric SanJuan, Gerardo Sierra, Juan-Manuel Torres-Moreno, and Jorge Vivaldi. Automatic specialized vs. non-specialized sentence differentiation. In *12th CICLING 2011, Tokyo, Japan*, pages 266–276, 2011.
- [6] Kees van Deemter, Mariët Theune, and Emiel Krahmer. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24, 2005.
- [7] Li Deng and Dong Yu. Deep learning: Methods and applications. *Foundations and Trends® in Signal Processing*, 7:197–387, 2014.
- [8] A. Edalat. Self-attachment: A holistic approach to computational psychiatry. In P. Érdi, B. Sen Bhattacharya, and A. Cochran, editors, *Computational Neurology and Psychiatry. Springer Series in Bio-/Neuroinformatics*, volume 6, pages 273–314. Springer, Cham, 2017.
- [9] Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. Learning semantic hierarchies via word embeddings. In *52nd Annual Meeting of the ACL (vol. 1: Long Papers)*, pages 1199–1209, 2014.
- [10] Hugo Gonçalo Oliveira. Poetryme: a versatile platform for poetry generation. In *Computational Creativity, Concept Invention and General Intelligence*, volume 1, Osnabrück, Germany, 2012. Institute of Cognitive Science.
- [11] Hugo Gonçalo Oliveira and Cardoso A. Poetry generation with poetryme. In *Computational Creativity Research: Towards Creative Machines*, volume 7, Paris, 2015. Atlantis Thinking Machines.
- [12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.
- [13] Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. Improving word representations via global context and multiple word prototypes. In *50th Annual Meeting of the ACL*, volume 1, pages 873–882. ACL, 2012.
- [14] Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. *arXiv preprint arXiv:1603.07771*, 2016.
- [15] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.
- [16] Susan Mcroy, Songsak Channarukul, and Syed Ali. An augmented template-based approach to text realization. *Natural Language Engineering*, 9:381 – 420, 12 2003.
- [17] T. Mikolov and G. Zweig. Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239, Dec 2012.
- [18] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *North American Chapter of the ACL: Human Language Technologies*, pages 746–751, 2013.
- [19] Paul Molins and Guy Lapalme. Jsrealb: A bilingual text realizer for web programming. In *15th European Workshop on Natural Language Generation (ENLG)*, pages 109–111, 2015.
- [20] N. Montfort. Through the park, 2008b.
- [21] N. Montfort. The two, 2008c.
- [22] N. Montfort. Taroko gorge, 2009.
- [23] Lluís Padró and Evgeny Stanilovsky. Freeling 3.0: Towards wider multilinguality. In *LREC2012*, 2012.
- [24] R.P. Pérez. *Creatividad Computacional*. Elibro Catedra. Larousse - Grupo Editorial Patria, 2015.
- [25] Mark O. Riedl and R. Michael Young. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing*, 24(3):303–323, Sep 2006.
- [26] Mike Sharples. An account of writing as creative design. *The Science of Writing*, pages 127–148, 1996.
- [27] Maheen Siddiqui, Roseli S. Wedemann, and Henrik Jeldtoft Jensen. Avalanches and generalized memory associativity in a network model for conscious and unconscious mental functioning. *Physica A: Statistical Mechanics and its Applications*, 490:127–138, 2018.

- [28] Gerardo Sierra. *Introducción a los Corpus Lingüísticos*. UNAM Mexico., 2018.
- [29] Giriprasad Sridhara, Emily Hill, Divya Muppaneni, Lori Pollock, and K. Vijay-Shanker. Towards automatically generating summary comments for java methods. In *IEEE/ACM, ASE '10*, pages 43–52, New York, NY, USA, 2010. ACM.
- [30] Grzegorz Szymanski and Zygmunt Ciota. Hidden markov models suitable for text generation. In *WSEAS International Conference on Signal, Speech and Image Processing (WSEAS ICOSSIP 2002)*, pages 3081–3084, 2002.
- [31] Juan-Manuel Torres-Moreno. Beyond stemming and lemmatization: Ultra-stemming to improve automatic text summarization. *CoRR*, abs/1209.3126, 2012.
- [32] Juan-Manuel Torres-Moreno. *Automatic Text Summarization*. Wiley, London, 2014.
- [33] Ke Wang and Xiaojun Wan. Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*, pages 4446–4452, 2018.
- [34] Roseli S. Wedemann and Luiz Alfredo Vidal de Carvalho. Some things psychopathologies can tell us about consciousness. In Alessandro E. P. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, editors, *ICANN 2012*, volume 7552, pages 379–386. Springer, Heidelberg, 2012.
- [35] Roseli S. Wedemann and Angel Ricardo Plastino. Física estadística, redes neuronales y freud. *Revista Núcleos*, 2, 2016.
- [36] Sean Welleck, Kianté Brantley, Hal Daumé III, and Kyunghyun Cho. Non-monotonic sequential text generation. *arXiv preprint arXiv:1902.02192*, 2019.
- [37] Xingxing Zhang and Mirella Lapata. Chinese poetry generation with recurrent neural networks. In *EMNLP*, pages 670–680, 01 2014.