Procesamiento del Lenguaje Natural, Revista nº 65, septiembre de 2020, pp. 29-36 recibido 30-04-2020 revisado 26-05-2020 aceptado 26-05-2020

Generación de frases literarias: un experimento preliminar

Generation of Literary Sentences: a Preliminary Approach

Luis-Gil Moreno-Jiménez[1,4], **Juan-Manuel Torres-Moreno**[1,3], **Roseli S. Wedemann**[2]
[1]Université d'Avignon/LIA
[2]Universidade do Estado do Rio de Janeiro
[3]Polytechnique Montréal
[4]Universidad Tecnológica de la Selva
luis-gil.moreno-jimenez@alumni.univ-avignon.fr,
juan-manuel.torres@univ-avignon.fr, roseli@ime.uerj.br

Resumen: En este trabajo abordamos la generación automática de frases literarias en español. Proponemos un modelo de generación textual basado en algoritmos estadísticos y análisis sintáctico superficial. Presentamos resultados preliminares que son bastante alentadores.
Palabras clave: Generación de texto, Modelos de lenguaje, Word embedding

Abstract: In this paper we address the automatic generation of literary phrases in Spanish. We propose a model for text generation based on statistical algorithms and Shallow Parsing. We present preliminary results that are quite encouraging.
Keywords: Natural Language Generation, Language Models, Word Embedding

1 Introducción

La Generación Automática de Texto (GAT), es un área del Procesamiento de Lenguaje Natural (PLN), que en los últimos años ha logrado avances importantes, bajo la convicción de desarrollar modelos computacionales capaces de simular cómo las personas manipulan el lenguaje. La mayoría de estos trabajos persiguen la automatización de ciertos procesos para aumentar la productividad en el ámbito industrial, académico, tecnológico, etc. (Szymanski y Ciota, 2002; Sridhara et al., 2010; Fu et al., 2014).

Sin embargo, existe un enfoque dentro de GAT que ha sido poco abordado: la generación automática de texto literario. Desde hace algún tiempo, diversos investigadores han trabajado en modelos generativos de este tipo, la mayor parte se avocan directamente a poemas, poesías o narrativas (Lebret, Grangier, y Auli, 2016; Brantley et al., 2019). Consideramos que la literatura, como "Proceso Creativo" Boden (2004), es el resultado que las personas desarrollan como parte de un proceso cognitivo complejo, y que la mejor manera de abordarla, es partir de conceptos más amplios como: emociones, asociación de conceptos (semántica) y estilos de redacción. Pensamos que, combinados adecuadamente, estos conceptos podrían usarse para modelar computacionalmente una parte del proceso creativo que una persona sigue para la generación de literatura.

La complejidad para automatizar la generación literaria reside en el análisis de los textos literarios, que sistemáticamente han sido dejados a un lado por varias razones. En primer lugar, el nivel de discurso literario es más complejo que el de los otros géneros. En segundo lugar, a menudo, los documentos literarios hacen referencia a mundos o situaciones imaginarias o alegóricas, a diferencia de géneros como el peridístico o enciclopédico que describen mayoritariamente 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.

Este proceso automatizado da lugar a un campo de investigación denominado Creatividad Computacional (CC) (Pérez y Pérez, 2015), donde el objetivo es modelar el "proce-

ISSN 1135-5948. DOI 10.26342/2020-65-3

so creativo" en un lenguaje que sea interpretable y reproducible en el dominio de lo calculable. Una gran variedad de modelos de IA han sido adaptados e incluso mejorados para lograr simular el proceso creativo a través de modelos computacionales (Colton, 2012). Debe considerarse que el objetivo principal de la CC no es solucionar problemas específicos en el ámbito industrial o académico, sino proponer nuevos paradigmas para la creación de obras artísticas (Colton, Wiggins, y others, 2012).

Otro problema es la falta de una definición universal de literatura; por lo que diversas definiciones son encontradas. Esto complica la tarea de evaluación, ya que, para lograr una percepción literaria homogénea, se debería partir de la misma definición de literatura. En este trabajo optaremos por introducir una definición pragmática de frase literaria, que servirá para nuestros modelos y experimentos.

Definición. *Una frase literaria se caracteriza por poseer elementos (sustantivos, verbos, adjetivos y adverbios) que son percibidos como elegantes y menos coloquiales que sus equivalentes en lengua general.*

Por ejemplo, la frase en lengua general:

- *"Me detuve a descansar luego de haber caminado mucho."*

puede ser ligeramente re-escrita para generar una frase literaria según nuestra definición:

- *"Tomé unos instantes para respirar y oxigenar mis pulmones... pues mi andar se había prolongado largo tiempo."*

En particular, proponemos crear artificialmente frases literarias utilizando modelos generativos y aproximaciones semánticas basados en corpora de lengua literaria. Buscamos, através de la combinación de estos elementos, 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, tampoco las mismas palabras, pero guarda la misma estructura sintáctica. En este artículo estudiamos el problema de la generación de texto literario en forma de frases aisladas, no a nivel de párrafos. La generación de párrafos puede ser objeto de trabajos futuros. Un protocolo de evaluación de la calidad de las frases generadas será presentado.

Este artículo está estructurado como sigue. La Sección 2 presenta un estado del arte de la creatividad computacional. La Sección 3 describe los corpora utilizados en nuestros experimentos. Los modelos usados son descritos en la Sección 4. Los resultados se encuentran en la Sección 5, antes de concluir en Sección 6 con algunas ideas de trabajos futuros.

2 Estado del arte

En la GAT se encuentran algunos trabajos con diversos objetivos. Szymanski y Ciota (2002) han desarrollado un modelo basado en cadenas de Markov para la generación de texto en polaco. El proceso inicia con un término dado por el usuario (estado inicial). Un proceso probabilístico calcula los estados siguientes. Cada estado es representado por n-gramas de letras o de palabras. El método demostró un mejor comportamiento, generando palabras de hasta 4 o 5 letras, considerando que en polaco esta es la longitud media de palabras.

Sridhara et al. (2010) utilizan un enfoque distinto. Presentan un algoritmo generativo de comentarios descriptivos aplicados a bloques de código en lenguaje Java. Se consideran algunas variables lingüísticas como los nombres de métodos, funciones e instancias. Estos elementos son procesados heurísticamente para generar texto descriptivo.

También existen trabajos menos extensos pero más precisos. Huang et al. (2012) proponen un modelo basado en redes neuronales para la generación de subconjuntos multipalabras. Este mismo objetivo se considera en (Fu et al., 2014), en donde se busca establecer y detectar la relación hiperónimo-hipónimo usando un modelo Word2vec (también basado en redes neuronales (Mikolov, Yih, y Zweig, 2013)). Los autores reportan una precisión de 0.70, al ser evaluado sobre un corpus manualmente etiquetado.

En cuanto a los trabajos relacionados a la GAT, se percibe una diferencia entre aquellos orientados a la generación literaria y aquellos que buscan la generación de texto en lengua general. Por ejemplo, en (Zhang y Lapata, 2014) se propone un modelo para la generación de poemas basado en dos premisas básicas: *¿qué decir?* y *¿cómo decirlo?* El modelo considera una lista de palabras clave para seleccionar un conjunto de frases. Estas frases son procesadas con una red neuronal (Mikolov y Zweig, 2012) para construir nuevas

combinaciones y formular nuevos contextos. 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.

Otros trabajos, como los de Oliveira (2012; Oliveira y Cardoso (2015) proponen modelos para la generación de poemas, basados en plantillas lingüísticas. Se utilizan listas de palabras clave para controlar el contexto bajo el cual los poemas son generados. Estos trabajos utilizan la herramienta PEN[1] para obtener la información gramatical de las palabras y crear nuevas plantillas. La ventaja de utilizar métodos basados en plantillas es que ayudan a mantener la coherencia y la gramaticalidad de los textos generados.

Modelos enfocados a la generación de poesía pueden ser analizados en los trabajos de Oliveira (2017) y Agirrezabal et al. (2013). Este último presenta un modelo estocástico, donde se calcula la probabilidad de aparición de etiquetas POS (*Part-of-Speech*), considerando las ocurrencias de estas etiquetas extraídas de diversos corpora. Después se generan nuevas secuencias y posteriormente se procede a la sustitución de las etiquetas POS de sustantivos y adjetivos.

3 Corpora utilizados

En esta sección, describimos los corpora utilizados en nuestros modelos para los experimentos. Se trata del corpus 5KL y del corpus 8KF, ambos creados en idioma español.

3.1 Corpus 5KL

Este corpus fue constituido con aproximadamente 5 000 documentos (en su mayor parte libros) en español. Los textos, en su mayoría, corresponden a géneros literarios: narrativa, poesía, teatro, ensayos, etc[2]. Los documentos originales, en formatos muy heterogéneos[3], fueron procesados para crear un único documento codificado en *utf8*. Dada su heterogeneidad, este corpus presenta una gran cantidad de errores: palabras cortadas o pegadas, símbolos, números, disposición no convencional de párrafos, etc. Lo que complica la tarea de análisis. Estas son, sin embargo, las condiciones que presenta un corpus literario real.

Las herramientas clásicas como FreeLing tienen mucha dificultad en tratar este tipo de documentos. Por ello, decidimos construir un segmentador de frases ad hoc para este tipo de corpus ruidoso. Las frases fueron segmentadas automáticamente, usando un programa en PERL 5.0 y expresiones regulares, para obtener una frase por línea.

Las características del corpus 5KL se encuentran en la Tabla 1[4]. Este corpus es empleado para entrenar el modelo Word2vec (Sección 4).

Corpus	**Frases**	**Tokens**
5KL	9 M	149 M
Media por doc.	2.4 K	37.3 K

Tabla 1: Corpus 5KL: 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, que a menudo otorgan una fluidez a la lectura y proporcionan los enlaces necesarios a las ideas.

3.2 Corpus 8KF

Decidimos crear un pequeño corpus controlado, exclusivamente compuesto de "frases literarias", que será utilizado para generar las plantillas o estructuras gramaticales (Sección 4.1). Este corpus 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) y demasiado largas ($N \geq 30$ palabras). Algunos elementos que sirvieron para seleccionar manualmente las frases "literarias" fueron: un vocabulario complejo y estético, el cual rara vez es empleado en el lenguaje común, además de la identificación de ciertas figuras literarias como la rima, la anáfora, la metáfora y otras. Las características del corpus 8KF se muestran en la Tabla 2.

4 Modelo de Generación Textual

A continuación, se describen las dos fases que conforman nuestro modelo. La primera

[1] http://code.google.com/p/pen

[2] Dada la dimensión de este corpus, no nos fue posible cuantificar los géneros manualmente. Una aproximación automática podrá realizarse a futuro.

[3] pdf, txt, html, doc, docx, odt, etc.

[4] M = 10^6 y K = 10^3.

Corpus	Frases	Tokens
8KF	7 679	114 K
Media por frase	–	15

Tabla 2: Corpus 8KF: frases literarias

consiste en la generación de una estructura gramatical que hemos denominado *estructura gramatical parcialmente vacía* (EGP), la cual se compone de palabras funcionales (conectores, verbos auxiliares, artículos, etc.) y etiquetas POS (*Part-of-Speech*)[5]. Estas últimas son obtenidas a través de un análisis morfosintáctico realizado con FreeLing [6] (Padró y Stanilovsky, 2012).

La segunda, consiste en la sustitución de las etiquetas POS con palabras semánticamente relacionadas a un contexto (*query*, Q), que es solicitado al usuario. Para ello, se emplea un modelo Word2vec (Mikolov, Yih, y Zweig, 2013) con interpretaciones geométricas.

4.1 Generación de EGP basado en texto enlatado

Esta técnica conocida como *texto enlatado* (*Canned Text*) (Molins y Lapalme, 2015) cuenta con la ventaja de agilizar el análisis sintáctico y permitir centrarnos directamente en el vocabulario a emplear (McRoy, Channarukul, y Ali, 2003; van Deemter, Theune, y Krahmer, 2005). La EGP es generada usando una frase del corpus 8KF (Sección 3), donde se reemplazan únicamente verbos, sustantivos o adjetivos $\{V, S, A\}$, por sus respectivas etiquetas POS. Las otras entidades lingüísticas, en particular las palabras funcionales, son conservadas.

El proceso inicia con la selección aleatoria de una frase original $f_o \in$ corpus 8KF de longitud $|f_o| = N$. Se selecciona una f_o para cada frase que se desee generar. 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 lingüísticos son los que mayor información aportan en un texto, independientemente de su género (Bracewell, Ren, y Kuriowa, 2005).

[5]Etiquetas que aportan información gramatical de cada palabra http://blade10.cs.upc.edu/freeling-old/doc/tagsets/tagset-es.html

[6]Desarrollado en el centro TALP (Universidad Politécnica de Cataluña) y puede ser obtenido en la dirección: http://nlp.lsi.upc.edu/freeling

Según nuestra hipótesis, al sustituirlos, lograremos la generación de frases nuevas por homosintaxis: semántica diferente, misma estructura[7].

La arquitectura del modelo se ilustra en la Figura 1. Los cuadros llenos representan palabras funcionales y los cuadros vacíos etiquetas POS a ser reemplazadas.

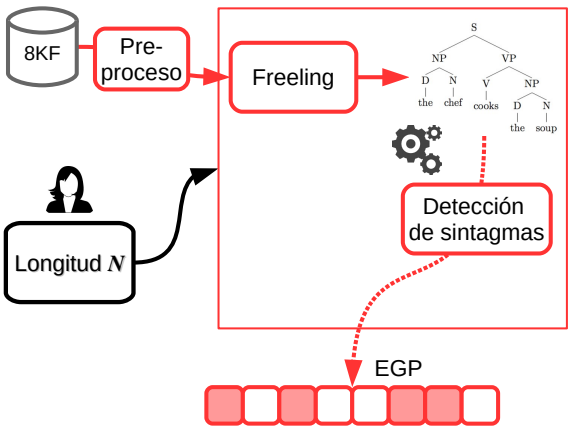

Figura 1: Modelo generativo *canned text*

4.2 Reemplazo de etiquetas POS

En esta fase, empleamos el modelo de aproximación semántica *Word2vec*, que utiliza un algoritmo basado en redes neuronales. El objetivo es obtener la representación vectorial de las palabras (*embeddings*) del corpus 5KL en un espacio n-dimensional[8], y poder calcular las distancias entre estas. El entrenamiento del modelo Word2vec se describe a continuación.

4.2.1 Modelo Word2vec

Para el entrenamiento y la implementación de Word2vec se utiliza la biblioteca Gensim[9], una implementación en Python de Word2vec[10]. El corpus 5KL es pre-procesado para uniformizar el formato del texto, eliminando caracteres que no son importantes en los análisis de PLN (como puntuación, números, etc.) (Torres-Moreno, 2014).

Se consideraron palabras con más de 5 ocurrencias en el corpus. Se definió una longitud de 10 palabras para la ventana contextual. Para las representaciones vectoriales se

[7]Al contrario de la paráfrasis que busca conservar la semántica, alterando la estructura sintáctica.

[8]Word2vec pertenece a un amplio campo de investigación dentro de PLN conocido como *Representation Learning* (Bengio, Courville, y Vincent, 2013).

[9]Disponible en: https://pypi.org/project/gensim/

[10]https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa

consideraron vectores de 60 dimensiones. El modelo Word2vec empleado en este trabajo es *continuous skip-gram model (Skip-gram)* (Mikolov et al., 2013).

Con el modelo entrenado, es posible obtener un conjunto de palabras, o *embeddings*, asociadas a una entrada definida por un *query* Q. Word2vec recibe un término Q y devuelve un léxico $L(Q) = (w_1, w_2, ..., w_m)$, que representa un conjunto de $m = 10$ palabras semánticamente próximas a Q. El valor de m fue definido de esta manera ya que se percibió que, mientras más se extiende el número de palabras proximas a Q, estas pierden más su relación con respecto a Q. Formalmente, representamos Word2vec: $Q \rightarrow L(Q)$.

4.2.2 Interpretación geométrica

En esta parte, determinamos las palabras más adecuadas que sustituirán las etiquetas POS de una EGP y así generar una nueva frase. Para cada etiqueta POS_k, $k = 1, 2, ...$ $\in$ EGP, que se desea sustituir, usamos el algoritmo descrito a continuación.

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 entradas de pares POS_k y una lista de palabras asociadas. Formalmente, se reprenta $\text{POS}_k \rightarrow V_k = \{v_{k,1}, v_{k,2}, ..., v_{k,i}\}$.

Luego se construye un vector para cada una de las tres palabras siguientes.

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

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

$$\vec{U} = (u_1, ..., u_{10}, u_{11}, ..., u_{20}, u_{21}, ..., u_{30})$$

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

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

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

$$\begin{aligned} o : \vec{X} &= (x_1, ..., x_{30}) \\ Q : \vec{Q} &= (q_1, .., q_{30}) \\ w : \vec{W} &= (w_1, ..., w_{30}) \end{aligned}$$

donde los valores de $\vec{X}$ son obtenidos tomando la distancia entre la palabra o y cada palabra $u_j \in \vec{U}, j = 1, ..., 30$. La distancia, $x_j = dist(o, u_j)$ es recuperada de Word2vec, donde $x_j \in [0, 1]$. 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 $j = 11, ..., 20$ y la palabra candidata w estará más próxima a las palabras u_j en las posiciones $j = 21, ..., 30$.

Enseguida, se calculan las similitudes coseno entre $\vec{Q}$ y $\vec{W}$ (1) y entre $\vec{X}$ y $\vec{W}$ (2),

$$\theta = \cos(\vec{Q}, \vec{W}) = \frac{\vec{Q} \cdot \vec{W}}{|\vec{Q}||\vec{W}|} \quad (1)$$

$$\beta = \cos(\vec{X}, \vec{W}) = \frac{\vec{X} \cdot \vec{W}}{|\vec{X}||\vec{W}|} \quad (2)$$

Estos valores de θ y β están normalizados en [0,1]. Se itera el proceso para todas las palabras del léxico $w \in V_k$. 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 β_i, $i = 1, ..., m$, 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

$$S_i = \left(\frac{\langle\theta\rangle}{\theta_i}\right) \cdot \left(\frac{\beta_i}{\langle\beta\rangle}\right) \tag{3}$$

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 (ver Figura 2).

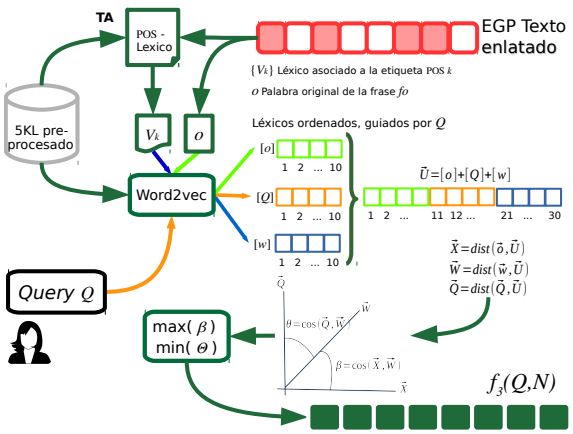

Figura 2: Aproximación semántica basada en interpretación geométrica min-max

5 *Resultados y evaluación*

Dados las características de nuestro modelo (idioma, corpora empleados, generación inspirada en homosintaxis), no es posible compararse con otros, ya que estos se centran en áreas específicas de la literatura y no parten de un enfoque general como es el caso en este trabajo. Sin embargo, previo a este modelo, realizamos dos experimentos basados en métodos estocásticos, empleando Cadenas de Markov, y un integración de *Canned Text* y Word2vec, con un enfoque simplificado. Los resultados de esos experimentos se detallan en (Moreno-Jiménez, Torres-Moreno, y Wedemann, 2020), pero a grosso modo los criterios evaluados así como la escala empleada fueron los mismos que se aplicaron en este trabajo. Para el modelo basado en Markov, se obtuvieron los siguientes resultados: gramaticalidad=0,55, coherencia=0,25 y contexto=0,67. El experimento integrado por *Canned Text* y Word2vec obtuvo: gramaticalidad=0,74, coherencia=0,56 y contexto=0,35.

Los resultados de esos experimentos nos permitieron detectar algunas deficiencias en los modelos iniciales, y proponer el modelo actual, cuyos resultados se muestran a continuación.

5.1 Resultados

Presentamos algunos ejemplos de frases generadas por nuestro modelo. Para el *query* Q, y una longitud en número de palabras N, los resultados se muestran en el formato $f(Q, N) =$ frase generada.

1. $f($`AMOR,10`$) =$ `En el aprecio está el cariño forzoso de una simpatía.`
2. $f($`AMOR,10`$) =$ `Los cariños no conocen de nada a un respeto loco.`
3. $f($`AMOR,10`$) =$ `No está la simpatía en las bondades de la envidia.`
4. $f($`GUERRA,9`$) =$ `Existe demasiada innovacion en torno a muy pocos sucesos.`
5. $f($`GUERRA,9`$) =$ `En la pelea todo debe motivo, menos la retirada.`
6. $f($`GUERRA,10`$) =$ `La codicia, siempre adversa, es terrible engendrada contra un desgraciado.`
7. $f($`SOL,9`$) =$ `Si tus dulces fueran amanecer, mis ojos marchitas fueran.`
8. $f($`SOL,11`$) =$ `Con rapidez, los monógamos impedimentos buscan para iluminar nos la luz.`
9. $f($`SOL,10`$) =$ `Incluso los luceros ingratos son comilones, y por tanto antiguos.`

5.2 Evaluación

Presentamos en esta sección un protocolo de evaluación manual. El experimento completo consistió en la generación de 45 frases. Se consideraron tres *queries*, $Q = \{$`AMOR, GUERRA, SOL`$\}$ y se generaron 15 frases de cada uno. Las frases fueron mezcladas antes de presentarlas a los 7 evaluadores seleccionados. Los evaluadores se eligieron considerando que poseen estudios universitarios y son hispanohablantes nativos, además de cierto hábito a la lectura que les permita la buena compresión de este tipo de textos. Se les pidió evaluar, usando la escala 0=mal, 1=aceptable y 2=correcto, los criterios siguientes:

- **Gramaticalidad:** ortografía, conjugaciones correctas, concordancia en género y número.
- **Coherencia:** legibilidad, percepción de una idea general.
- **Contexto:** relación de la frase con respecto al *query*.

Se realizó una adaptación del Turing. A los evaluadores se les hizo creer que había algunas frases escritas por personas y otras escritas por los algoritmos. Se les pidió indicar cuáles frases pensaban que habían sido generadas por personas (0) y cuáles por algoritmos (1). La Tabla 3 presenta la media aritmética y la desviación estándar de los resultados obtenidos normalizados a una escala entre 0-1.

Criterio	Resultados
Gramaticalidad	**0.77 ± 0.13**
Coherencia	**0.60 ± 0.14**
Contexto	**0.53 ± 0.19**
Turing	**0.44 ± 0.15**

Tabla 3: Evaluación del sistema

Se observa que las frases son percibidas como gramaticales y coherentes, con una evaluación de 0,77 y 0,60 respectivamente. El contexto obtuvo un resultado más bajo. Esto puede deberse a que las EGP contienen elementos fijos (palabras funcionales) que en ocasiones pueden alterar el contexto o semántica de las palabras insertadas. A pesar de ello, los resultados desde una perspectiva general son bastante alentadores.

En el test de Turing, los evaluadores perciben un 44 % como frases generadas por una persona. Esto, aunque parece ser un score bajo, comparado con los trabajos relacionados con este tema, resulta ser una buena evaluación, considerando que el objetivo no es generar texto aleatorio, sino un texto con características literarias.

6 Conclusiones

En este trabajo presentamos una primera aproximación de un modelo generativo de texto literario usando modelos neuronales de tipo Word embedding. El modelo produce frases aisladas en español con un cierto contenido literario. No se requiere prácticamente intervención del usuario (excepto por el query y la longitud de la frase requerida). La generación de párrafos y el uso de rimas sera el objeto de trabajos futuros (Medina-Urrea y Torres-Moreno, 2019). Dada la estructura del modelo propuesto, la extensión a otros idiomas (francés y portugués) también será contemplada (Charton y Torres-Moreno, 2011).

Bibliografía

Agirrezabal, M., B. Arrieta, A. Astigarraga, y M. Hulden. 2013. Pos-tag based poetry generation with wordnet. En *14th European Workshop on Natural Language Generation*, páginas 162–166. ACL.

Bengio, Y., A. Courville, y P. Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Boden, M. A. 2004. *The creative mind: Myths and Mechanisms*. Routledge.

Bracewell, D., F. Ren, y S. Kuriowa. 2005. Multilingual single document keyword extraction for information retrieval. En *2005 International Conference on Natural Language Processing and Knowledge Engineering*, páginas 517–522, Wuhan, China. IEEE.

Brantley, K., K. Cho, H. Daumé, y S. Welleck. 2019. Non-monotonic sequential text generation. En *2019 Workshop on Widening NLP*, páginas 57–59, Florence, Italy. ACL.

Charton, E. y J.-M. Torres-Moreno. 2011. Automatic modeling of logical connectors by statistical analysis of context. *Canadian Journal of Information and Library Science*, 35(3):287–306.

Colton, S. 2012. *Automated theory formation in pure mathematics*. Springer Science & Business Media.

Colton, S., G. A. Wiggins, y others. 2012. Computational creativity: The final frontier? En *20th European Conference on Artificial Intelligence*, páginas 21–26. ACL.

Fu, R., J. Guo, B. Qin, W. Che, H. Wang, y T. Liu. 2014. Learning semantic hierarchies via word embeddings. En *52nd Annual Meeting of the ACL*, volumen 1, páginas 1199–1209, Baltimore, Maryland, USA. ACL.

Huang, E. H., R. Socher, C. D. Manning, y A. Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. En *50th Annual Meeting of the ACL*, volumen 1, página 873–882, USA. ACL.

Lebret, R., D. Grangier, y M. Auli. 2016. Neural text generation from structured data with application to the biography domain. *arXiv*, página arXiv 1603.07771.

McRoy, S., S. Channarukul, y S. Ali. 2003. An augmented template-based approach to text realization. *Natural Language Engineering*, 9:381 – 420.

Medina-Urrea, A. y J.-M. Torres-Moreno. 2019. Rimax: Ranking semantic rhymes by calculating definition similarity.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, y J. Dean. 2013. Distributed representations of words and phrases and their compositionality. En *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., páginas 3111–3119.

Mikolov, T., W.-t. Yih, y G. Zweig. 2013. Linguistic regularities in continuous space word representations. En *NACACL: Human Language Technologies*, páginas 746–751, Atlanta, Georgia, USA. ACL.

Mikolov, T. y G. Zweig. 2012. Context dependent recurrent neural network language model. En *2012 IEEE Spoken Language Technology Workshop (SLT)*, páginas 234–239, Miami, FL, USA. IEEE.

Molins, P. y G. Lapalme. 2015. JSrealB: A bilingual text realizer for web programming. En *15th ENLG*, páginas 109–111, Brighton, UK. ACL.

Moreno-Jiménez, L.-G., J.-M. Torres-Moreno, y R. S. Wedemann. 2020. Generación automática de frases literarias. *Linguamatica*. Accepted.

Oliveira, H. G. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. En *10th ICNLG*, páginas 11–20.

Oliveira, H. G. 2012. Poetryme: a versatile platform for poetry generation. En *Computational Creativity, Concept Invention and General Intelligence*, volumen 1, Osnabrück, Germany. Institute of Cognitive Science.

Oliveira, H. G. y A. Cardoso. 2015. Poetry generation with poetryme. En *Computational Creativity Research: Towards Creative Machines*, volumen 7, Paris, France. Atlantis Thinking Machines.

Padró, L. y E. Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. En *8th LREC)*, Istanbul, Turkey.

Pérez y Pérez, R. 2015. *Creatividad Computacional*. Larousse - Grupo Editorial Patria, México.

Sridhara, G., E. Hill, D. Muppaneni, L. Pollock, y K. Vijay-Shanker. 2010. Towards automatically generating summary comments for java methods. En *IEEE/ACM International Conference on Automated Software Engineering*, página 43–52, Antwerp, Belgium. ACM.

Szymanski, G. y Z. Ciota. 2002. Hidden markov models suitable for text generation. En N. Mastorakis V. Kluev, y D. Koruga, editores, *WSEAS*, páginas 3081–3084, Athens, Greece. WSEAS - Press.

Torres-Moreno, J.-M. 2014. *Automatic Text Summarization*. ISTE, Wiley, London, UK, Hoboken, USA.

van Deemter, K., M. Theune, y E. Krahmer. 2005. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24.

Zhang, X. y M. Lapata. 2014. Chinese poetry generation with recurrent neural networks. En *2014 EMNLP*, páginas 670–680, Doha, Qatar. ACL.