Propuesta recibida en octubre 2019 y aceptada para publicación en mayo 2020.

Generación automática de frases literarias

Automatic Generation of Literary Sentences

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

Eric SanJuan
Université d'Avignon/LIA
eric.sanjuan@univ-avignon.fr

Resumen

En este artículo abordamos el tema de la generación automática de frases literarias, que es una parte importante de los estudios relacionados al área de la Creatividad Computacional (CC). Proponemos tres modelos de generación textual guiados por un contexto, basados principalmente en algoritmos estadísticos y análisis sintáctico superficial. Los textos generados fueron evaluados por siete personas a partir de 4 criterios: gramaticalidad, coherencia, relación con el contexto y una adaptación del test de Turing, en donde se pidio a los evaluadores clasificar los textos en: textos generados automáticamente y textos generados por humanos. Los resultados obtenidos son bastante alentadores.

Palabras clave

corpora literarios, generación automática de frases, cadenas de Markov, Word2vec

Abstract

In this article, we regard the task of automatic generation of literary sentences, which is an important topic in the area of Computational Creativity. We propose three generative models mainly based on statistical algorithms and shallow parsing. The generated texts were evaluated by seven persons according to four criteria: grammar, coherence, context related, and an adaptation of the Turing test. We present preliminary results of their implementations that are quite encouraging.

Keywords

literary corpora, automatic sentence generation, Markov chains, Word2Vec

1. Introducción

Los investigadores en Procesamiento de Lenguaje Natural (PLN) durante mucho tiempo han utilizado diversos corpora constituidos por documentos enciclopédicos (principalmente Wikipedia), periodísticos (periódicos o revistas) o especializados (documentos legales, científicos o técnicos) para el desarrollo y pruebas de sus modelos (Torres-Moreno, 2014; Iria et al., 2011; Martínez, 2018).

El uso y análisis de los 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 Boden (2004), 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.

DOI: 10.21814/lm.12.1.308

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 campo de investigación, denominado Creatividad Computacional (CC) (Pérez y Pérez, 2015), en donde se retoman los conceptos: CT y la CE propuestos por Boden (2004). Es en este campo donde nosotros hemos trabajado para la generación de frases literarias.

Por otro lado, 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 1 *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.*

Por ejemplo, la frase en lengua general:

- *Me paré a ver unos libros viejos en la librería que está en la esquina de mi casa.*

Puede ser ligeramente re-escrita para generar tres frases literarias según nuestra definición:

- *Me detuve a mirar libros antiguos en la librería próxima a mi casa.*
- *Miré durante unos momentos algunos libros antiguos en la librería cercana a mi casa.*
- *Hojeé durante algunos instantes libros viejos en la librería cercana a mi hogar.*

Por supuesto, un autor puede decidir escribir un texto literario basado exclusivamente en frases de lengua general. Por ejemplo, José Agustín en "De perfil" donde el fragmento: *"...me quedé dormido en el Jardín. Supongo que el sol y lo fresco del aire crearon el término exacto para adormecerme."*[1], usa frases literarias dentro de un texto desbordante de lengua general. Sin embargo, nosotros no intentaremos mezclar ambas lenguas y nos restringiremos a analizar y generar frases literarias.

En particular, proponemos crear artificialmente frases literarias, utilizando modelos generativos y aproximaciones semánticas basados en corpora 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 generación automática de textos. En la Sección 3 describimos los corpora 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. Estado del arte

A continuación, se presenta un estudio del estado del arte en donde se mencionan trabajos para la generaciòn de texto con enfoques variados y objetivos bastante interesantes. En principio, mostramos algunos trabajos que no estàn relacionados a la CC literaria. Posteriormente, analizamos investigaciones dedicadas a la generaciòn textual dentro del marco de la CC, como generaciòn de poemas, poesìas y otras formas literarias.

Durante nuestra investigaciòn, hemos percibido que la CC no busca solucionar los problemas de la sociedad en sus variados aspectos, sino encontrar nuevos paradigmas para la creación de obras con un importante valor cultural y artístico (Colton & Wiggins, 2012). Sin embargo, 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).

2.1. Generación textual no literaria

La generación de texto es una tarea relativamente clásica, que ha sido estudiada en diversos trabajos. Por ejemplo, Szymanski & Ciota (2002) presentam 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,

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

[1] J. Agustín. *De perfil*, Joaquín Mortiz, México, 1993.

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 (Torres-Moreno, 2012).

También hay trabajos que realizan análisis más profundos para generar no solamente palabras, sino párrafos completos. Sridhara et al. (2010) 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 existen trabajos con un alcance más limitado pero de mayor precisión. Huang et al. (2012) proponen la evaluación de un conjunto de datos con un modelo basado en redes neuronales para la generación de subconjuntos de multipalabras. Este mismo análisis se considera por Fu et al. (2014), en donde se busca establecer o detectar la relación hiperónimo-hipónimo con la ayuda del modelo Word2vec, también basado en redes neuronales (Mikolov et al., 2013b). Esta propuesta reporta una precisión de 0.70, al ser evaluado sobre un corpus manualmente etiquetado.

2.2. Generación de poesía

También se encuentran trabajos de generación textual que se proponen como meta resultados con un valor literario. La generación de texto literario es un proceso distinto a la generación de texto general (Lebret et al., 2016; Welleck et al., 2019), y ha sido abordado desde los años 60's por investigadores del campo de humanidades, siendo hasta principios del año 2000 abordada fuertemente por la Ciencias Computacionales (Gonçalo Oliveira, 2017), Entre estos, se tienen trabajos para la generación de poesía o poemas.

Zhang & Lapata (2014) proponen un modelo para la generación de poemas que 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 (Mikolov & Zweig, 2012), 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.

Gonçalo Oliveira (2012); Gonçalo Oliveira & Cardoso (2015) proponen un modelo de generación de poemas basado en el uso 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 PEN[2] 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.

Otros trabajos han sido propuestos para la generación de poesía, como en (Agirrezabal et al., 2013), donde se presenta un método bastante interesante, en donde a partir del análisis de diversos corpora, se extraen las secuencias de etiquetas POS con sus respectivas inflexiones para calcular la probabilidad de aparición de cada una de ellas. Este método estocástico sirve para la generación de nuevas secuencias y posteriormente se procede a la sustitución de las etiquetas POS. Se realizaron tres experimentos para la sustitución. En el primero se sustituyen todas las etiquetas de las

[2]Disponible en: `http://code.google.com/p/pen`

secuencias POS por palabras que respeten la gramaticalidad de la etiqueta. En el segundo se sustituyen únicamente adjetivos y sustantivos bajo la misma condición, y finalmente en el tercer experimento sólo se reemplazan sustantivos con palabras con una relación semántica determinada.

2.3. Generación de narrativas

La literatura es una actividad artística que exige capacidades creativas importantes y que ha llamado la atención de científicos desde hace cierto tiempo. Diversos investigadores han trabajado en proyectos que permiten la generación de texto literario cruzando la frontera de textos cortos como poemas o poesía, para dar lugar a la generación de textos más extensos.

Riedl & Young (2006) presentan un conjunto de algoritmos para la generación de una guía narrativa basada en la idea de Creatividad Exploratoria (Boden, 2004). 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. Clark et al. (2018) proponen un modelo de generación de texto narrativo a partir del análisis de *entidades*. Dichas *entidades* son 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.

El modelo sentiGAN (Ke & Xiaojun, 2018) pretende generar texto con un contexto emocional. Se trata de una actualización del modelo GAN (*Generative Adversarial Net*) (Goodfellow et al., 2014) 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.

Pérez y Pérez (2015) presentam una revisión interesante del estado del arte en este tema, donde se mencionan algunos de los primeros intentos de generación automática de textos literarios. Por ejemplo, el modelo "Through the park" (Montfort, 2008b), 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" (Montfort, 2008c) y "Taroko Gorge" (Montfort, 2009) 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 Pérez y Pérez (2015). 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 and Reflexive)* (Sharples, 1996). 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.

Otro trabajo que contempla la generacion de narrativas es el que se presenta en (Gervás et al., 2015), en donde se expone el método bajo el cual, algunos escritores reutilizan las experiencias recabadas en textos leídos o escritos, estas son apli-

cadas para la creación de nuevos textos. En este trabajo, estas experiencias son traducidas como escenarios, estructuras, acciones, etc., que sirven como datos de entrenamiento para la generación de nuevas narrativas.

Nuestro modelo de generación de frases no fue concebido para la generación de poesía o narrativa. Mas bien está dentro de un cuadro general de generación automática, teniendo como objetivo la construcción de un generador artificial de texto literario. Desde este punto de vista es sólo un módulo de un sistema mas complejo en perspectiva, que contempla la semántica, el manejo de figuras literarias y las emociones.

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 mayoria, correspondem a los géneros literarios: narrativa, poesía, teatro, ensayos, etc[3]. Los documentos originales, en formatos muy heterogéneos[4], fueron procesados para crear un único documento codificado en *UTF-8*. Dada su heterogeneidad, este corpus presenta una gran cantidad de errores (palabras cortadas o pegadas, símbolos extraños y disposición no convencional de párrafos).

Las herramientas clásicas como FreeLing (Padró, 2012) tienen mucha dificultad en tratar estes tipos 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 y expresiones regulares, para obtener una frase por línea.

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

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.

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

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

Otra desventaja de este corpus es el ruido que contiene. Por lo que, 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

Decidimos crear un pequeño corpus controlado, exclusivamente compuesto de "frases literarias", que será utilizado en la fase generativa de los modelos propuestos. 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). 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. Algunos ejemplos de frases literarias son los siguientes:

- *La mentira y la verdad no pueden vivir en paz.*
- *El amor, como la tos, no puede ocultarse.*
- *Si tu belleza fuera enfermedad, vida mía, no habría remedio.*
- *Grabad esto en vuestro corazón: cada día es el mejor del año.*

Las características del corpus 8KF se muestran en el Cuadro 2. Este corpus fue utiliza-

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

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

[5]M representa un valor de 10^6 y K de 10^3.

do principalmente en los dos modelos generativos: modelo basado en cadenas de Markov (Sección 4.1.1) y modelo basado en la generación de Texto enlatado (*Canned Text*, Sección 4.1.2).

	Frases	Tokens	Caracteres
8KF	7 679	114 K	652 K
Media por frase	—	15	85

Cuadro 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*), combinados con tres modelos de aproximación semántica usando *Word2vec*.

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 (Padró, 2012) que permite análisis lingüísticos en varios idiomas[6] y que además de devolvernos las etiquetas POS, que nos permiten saber si la palabra analizada es un verbo, sustantivo, adjetivo, etc., también nos da información acerca de las inflexiones en ella, es decir, conjugaciones, género, número, etc. Por ejemplo, para la palabra "Profesor" FreeLing genera la etiqueta POS `[NCMS000]`. La primera letra indica un sustantivo (**N**oun), la segunda un sustantivo común (**C**ommon); la tercera indica el género masculino (**M**ale) y la cuarta da información de número (**S**ingular). Los 3 últimos caracteres dan información detallada del campo semántico, entidades nombradas, etc.[7] 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 una *Estructura gramatical parcialmente vacía* (EGP), compuesta de etiquetas POS y de palabras funcionales (artículos, pronombres, conjunciones, etc.).

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 (*query*) y una longitud $3 \leq N \leq 15$, dados por el usuario. Los corpora 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 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, utilizando redes neuronales con Word2vec. 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.

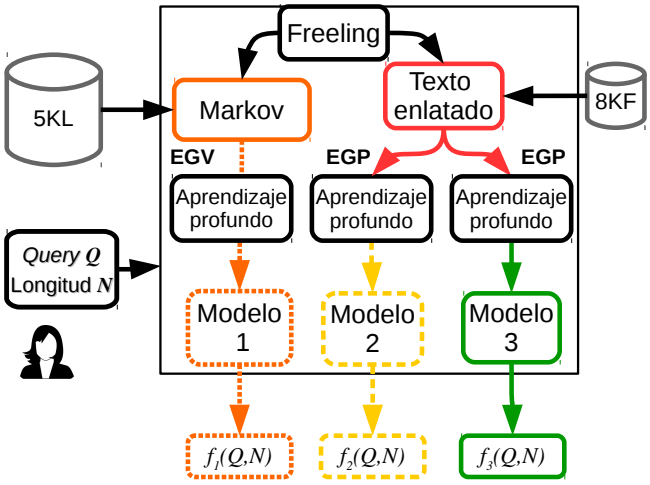

Figura 1: Arquitectura general de los modelos.

La Figura 1 muestra la arquitectura general de nuestro sistema. En la Sección 4.1 se describen los dos modelos generativos, y enseguida los tres modelos de aproximación semántica.

[6] FreeLing ha sido desarrollado en el centro TALP (Universidad Politécnica de Cataluña). Puede ser obtenido en la dirección: `http://nlp.lsi.upc.edu/freeling`

[7] Más detalles de las etiquetas FreeLing en `http://blade10.cs.upc.edu/freeling-old/doc/tagsets/tagset-es.html`

4.1. Modelos generativos

A continuación, se presentan dos modelos generativos de estructuras gramaticales en sus dos variantes, Estructuras Gramaticales Vacías (EGV) y Estructuras Gramaticiales Parcialmente Vacías (EGP), que sirven a los modelos descritos en las secciones 4.2, 4.3 y 4.4 para la generación de frases.

4.1.1. 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 (Manning & Schütze, 1999), 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 el modelo descrito en la Sección 4.2. La arquitectura general de este modelo se muestra en la Figura 2.

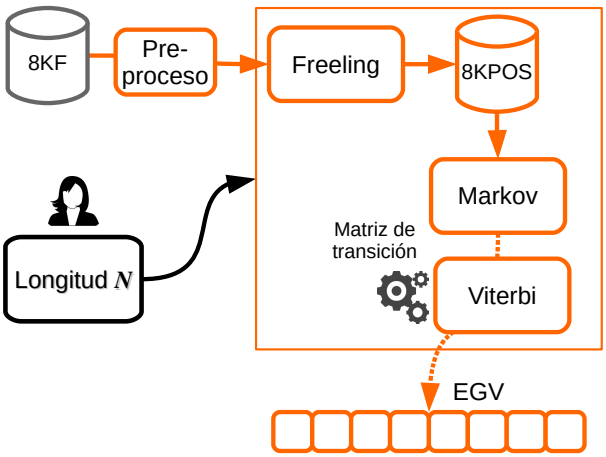

Figura 2: Modelo generativo estocástico (Markov) que produce una estructura gramatical vacía EGV.

4.1.2. 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 4.2). Decidimos entonces utilizar métodos de generación textual guiados por estructuras morfosintácticas fijas: el *texto enlatado*. Molins & Lapalme (2015) 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. McRoy et al. (2003); van Deemter et al. (2005) 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 enlatado*[8].

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 gra-

[8] http://projects.ict.usc.edu/nld/cs599s13/LectureNotes/cs599s13dialogue2-13-13.pdf

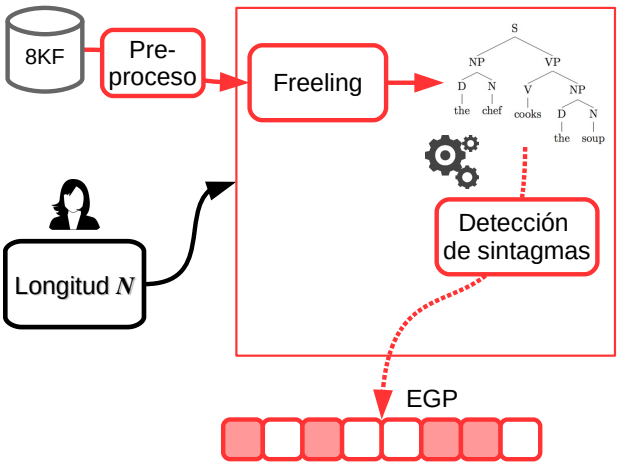

Figura 3: Modelo generativo de Texto enlatado que produce una estructura parcialmente vacía.

maticales 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 (Bracewell et al., 2005). Nuestra hipótesis es que al cambiar solamente estos elementos, simulamos la generación de frases por homosintaxis: semántica diferente, misma estructura[9].

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.

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

4.2. Modelo 1: Markov y Word2vec

En este modelo se retoman las estructuras gramaticales vacías (EGV), descritas en la Sección 4.1.1, 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 que combina el modelo generativo de Markov (Sección 4.1.1), con un algoritmo de aproximación semántica que utiliza un modelo de redes neuronales, el Word2vec. La labor de Word2vec es obtener la representación de una palabra en un espacio vectorial (*embeddings*), a través de un análisis contextual[10]. 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 Word2vec que utiliza una representación vectorial del corpus 5KL. Para este, utilizamos la biblioteca Gensim[11], una implementación en Python de Word2vec[12]. Con este algoritmo, se obtiene un conjunto de palabras, o *embeddings*, asociadas a un contexto definido por un *query* Q. Word2vec recibe un término Q y devuelve un léxico $L(Q) = (w_1, w_2, \ldots, w_m)$, que representa un conjunto de $m = 10$ palabras semánticamente próximas a Q. El valor de m fue definido de esta manera ya que se 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: $Q \rightarrow L(Q)$.

Para el entrenamiento de Word2vec se consideran palabras con más de 5 ocurrencias en el corpus. La ventana contextual definida tiene una dimensión de 10. Para las dimensiones de las representaciones vectoriales se hicieron pruebas dentro de un rango de 50 a 100, siendo 60 la dimensión con la que se obtuvieron embeddings mejores relacionados. El modelo de entrenamiento fue *continuous skip-gram model (Skip-gram)*, el cual funciona mejor con copora de tamaños significativos (Mikolov et al., 2013a).

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).

[10] Word2vec pertenece a un amplio campo de investigación dentro de PLN, conocido como *Representation Learning* (Bengio et al., 2013).

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

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

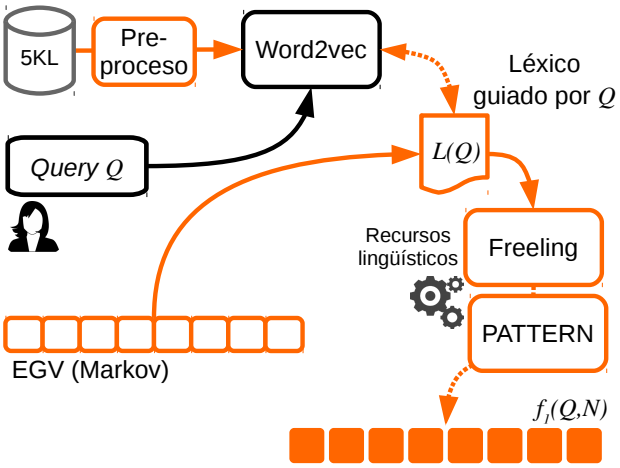

Figura 4: Modelo 1 de aproximación semántica, usando Markov y Word2vec.

Las etiquetas POS_Φ serán reemplazadas por palabras obtenidas de recursos lingüísticos (diccionarios) construídos con la ayuda de FreeLing. Los diccionarios consisten en entradas de pares: POS_Φ y una lista de palabras y signos asociados, formalmente $\text{POS}_\Phi \rightarrow l(\text{POS}_\Phi) = (l_1, l_2, \ldots, l_j)$. 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, $L(Q)$, producidas por Word2vec. Si ninguna de las palabras de $L(Q)$ tiene la forma sintáctica exigida por POS_λ, empleamos la biblioteca PATTERN[13], 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.

[13] https://www.clips.uantwerpen.be/pattern

4.3. Modelo 2: Texto enlatado, Word2vec y análisis morfosintáctico

En este modelo proponemos una combinación entre el modelo de *texto enlatado* (Sección 4.1.2) y el algoritmo Word2vec entrenado sobre el corpus 5KL. El objetivo es eliminar las iteraciones del Modelo 1, que son necesarias cuando las etiquetas POS[14] 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 entradas de pares POS_k y una lista de palabras asociadas. Formalmente, se reprenta $\text{POS}_k \rightarrow V_k = \{v_{k,1}, v_{k,2}, \ldots, v_{k,i}\}$. 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 $\text{POS}_k \in \text{EGP}$, $k = 1, 2, \ldots$, 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), $dist(Q, v_{k,i})$, entre cada palabra del vocabulario, $v_{k,i}$, y el *query* Q del usuario. 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.

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

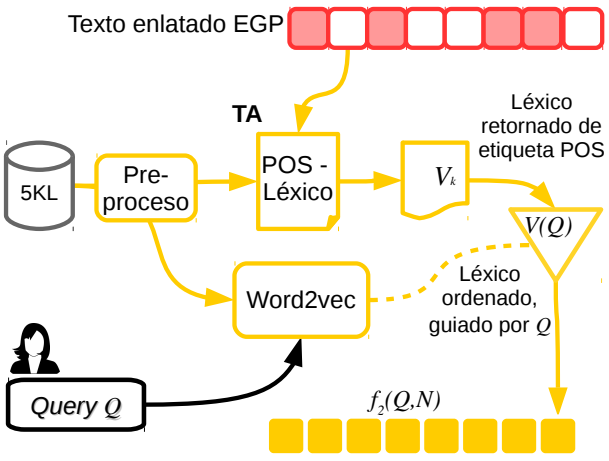

Figura 5: Modelo 2 de aproximación semántica basada en *Word2vec* y análisis morfosintáctico.

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.

4.4. Modelo 3: Texto enlatado, Word2vec e interpretación geométrica

El Modelo 3 reutiliza varios de los recursos anteriores: el algoritmo Word2vec, la Tabla Asociativa TA y la estructura gramatical 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, $k = 1, 2, \ldots \in$ 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 4.1.2), correspondiente a la etiqueta POS_k. Esta palabra permite recrear un contexto del cual la nueva frase debe alejarse, evitando producir una paráfrasis.

- Q: es la palabra que define al *query* proporcionado por el usuario.

- w: la 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.

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

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

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

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

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

$$o : \vec{X} = (x_1, \ldots, x_{10}, x_{11}, \ldots, x_{20}, x_{21}, \ldots, x_{30}),$$
$$Q : \vec{Q} = (q_1, \ldots, q_{10}, q_{11}, \ldots, q_{20}, q_{21}, \ldots, q_{30}),$$
$$w : \vec{W} = (w_1, \ldots, w_{10}, w_{11}, \ldots, w_{20}, w_{21}, \ldots, w_{30}),$$

donde los valores de $\vec{X}$ son obtenidos tomando la distancia entre la palabra o y cada palabra $u_j \in \vec{U}, j = 1, \ldots, 30$. 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 $j = 11, \ldots, 20$ y la palabra candidata w estará más próxima a las palabras u_j en las posiciones $j = 21, \ldots 30$.

Enseguida, se calculan las similitudes coseno entre $\vec{Q}$ y $\vec{W}$ (4) y entre $\vec{X}$ y $\vec{W}$ (5),

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

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

Estos valores de θ y β están normalizados en [0,1]. 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 β_i, $i = 1, \ldots, m$, 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

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

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.

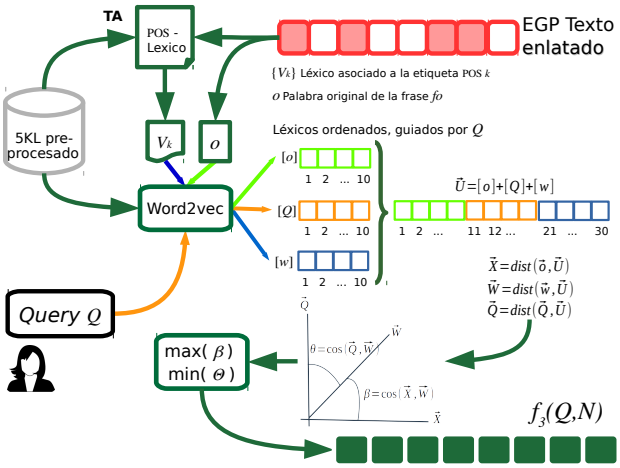

Figura 6: Modelo 3 de aproximación semántica, basada en interpretación geométrica min-max.

5. Experimentos y resultados

Se han diseñado tres experimentos para validar los tres modelos semánticos desarrollados en este trabajo. A partir de un *query* del usuario y de una longitud de palabras, el sistema realiza los procesos siguientes.

- **Modelo 1:** El modelo generativo de cadenas de Markov produce una EGV que se envía al modelo de aproximación semántica para generar las frases f_1.
- **Modelo 2**: El modelo generativo de *texto enlatado* produce una EGP y se envía al modelo de aproximación semántica para generar las frases f_2.
- **Modelo 3**: El modelo de *texto enlatado* produce una EGP y se utiliza el modelo de interpretación geométrica para la generación de las frases f_3.

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

Enseguida presentamos unos ejemplos de las frases obtenidas en función de los experimentos propuestos. Para el *query*, Q, y la longitud en número de palabras, N, los resultados se muestran en el formato,

$$f(Q, N) = \text{frase generada} . \qquad (7)$$

Los resultados han sido generados empleando los *queries* Q=`{GUERRA,SOL}` en todos los casos.

Modelo 1

1. $f(\texttt{GUERRA}, 12)$ = *El ejército conquista mediante el enemigo. La batalla es la guerra desde.*
2. $f(\texttt{GUERRA}, 13)$ = *Toda batalla en rebelión es la guerra contra el ejército en el combate.*
3. $f(\texttt{SOL}, 12)$ = *La luna salvo la lluvia sobre el ocaso hacia el cielo brilla.*
4. $f(\texttt{SOL}, 13)$ = *Cuántos naveguen salvo iluminar para el cielo hacia la aurora es la luna.*

Modelo 2

1. $f(\texttt{GUERRA}, 9)$ = *El incivil comportamiento para la magnificencia es la dicha.*
2. $f(\texttt{GUERRA}, 10)$ = *La cultura es la religión de dogmatizar los bienes caducos.*
3. $f(\texttt{SOL}, 11)$ = *Brilla que contener siempre. Nunca se es dominado de el todo.*
4. $f(\texttt{SOL}, 10)$ = *El rocío exhala el bosque después de haberlo fatigado.*

Modelo 3

1. $f(\texttt{GUERRA}, 9)$ = *Existe demasiada innovacion en torno a muy pocos sucesos.*
2. $f(\texttt{GUERRA}, 9)$ = *En la pelea todo debe motivo, menos la retirada.*

3. $f(\texttt{SOL}, 11)$ = *Con rapidez, los monógamos impedimentos buscan para iluminar nos la luz.*
4. $f(\texttt{SOL}, 10)$ = *Incluso los luceros ingratos son comilones, y por tanto antiguos.*

5.2. Protocolo de evaluación e interpretación de 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*, $Q = \{\texttt{AMOR, GUERRA, SOL}\}$, 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.
- **Coherencia:** legibilidad, percepción de una idea general.
- **Contexto:** relación de la frase con respecto al *query*.

En el Anexo A se muestran las frases generadas para la evaluación manual.

El mini-test de Turing fue evaluado con una nota de 0 o 1. 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).

Los resultados de la evaluación se presentan en la Figura 7, en la forma de gráfica de barras, donde cada barra representa un criterio evaluado. Los valores representados corresponden a la moda para cada uno de los criterios.

La primera sección de barras ilustra la evaluación de las frases generadas por el Modelo 1. En ella se puede apreciar como los evaluadores percibieron una estrecha relación con el contexto *query* (barra roja) y una gramática aceptable (barra gris). Sin embargo para este modelo, la barra de coherencia (barra azul) es nula, lo que indica que los evaluadores no perciben las frases como coherentes.

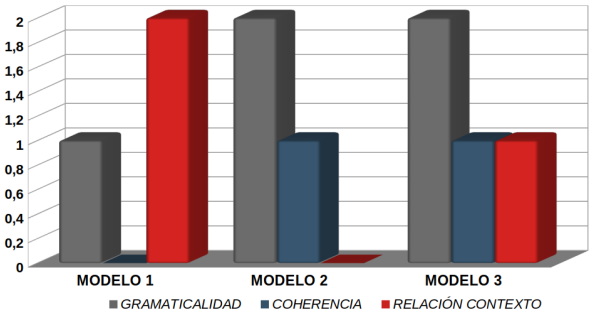

Figura 7: Evaluación de la coherencia, gramaticalida y contexto.

La estrecha relación con el contexto se debe al alto grado de libertad que caracteriza a la EGV generada por el modelo de Markov. Esta EGV permite que todos los elementos de la estructura puedan ser sustituidos por un léxico guiado por los resultados del algoritmo Word2vec.

Para los resultados del Modelo 2, los evaluadores perciben frases razonablemente coherentes y gramaticalmente correctas. Sin embargo, los evaluadores no percibieron una relación evidente entre el contexto de la frase generada y el *query*. Esto se debe a que las frases generadas reportan, en su mayoría, el mismo contexto o idea de la frase original, pudiendo ser interpretado como una paráfrasis elemental, que no es lo que deseamos.

Finalmente, el Modelo 3 genera frases coherentes, gramaticalmente correctas y más bien relacionadas al *query* que el Modelo 2, siendo el único modelo donde los tres criterios evaluados se hacen presentes. Esto se logra siguiendo una intuición opuesta a la paráfrasis: buscamos conservar la estructura sintáctica de la frase original, generando una semántica completamente diferente.

En general se puede apreciar que a diferencia de los modelos 1 y 2, en donde sólo 2 de cada 3 criterios fueron percibidos con claridad, el Modelo 3 es el único que obtuvo resultados positivos en los tres criterios.

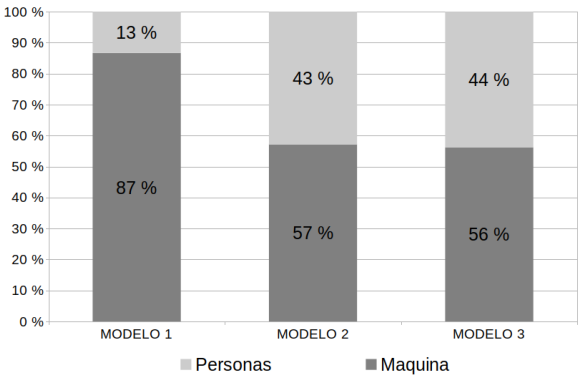

Figura 8: Evaluación del mini-test de Turing.

En la Figura 8, de acuerdo a los resultados del mini-test de Turing, se muestra en tonalidad oscura la moda con la que los evaluadores percibieron las frases como escritas por personas, mientras que en una tonalidad clara (gris) cuando las frases se percibieron como generadas por máquinas. En general, los evaluadores perciben con el 44 % de las frases generadas por el Modelo 3 como generadas por una persona. Esta es la mejor percepción de los tres modelos.

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*, Word2vec con redes neuronales y una interpretación del tipo IR.

Uno de los principales problemas detectados en los tres modelos es la pérdida de coherencia en las frases generadas. Esto puede deberse a la ambigüedad gramatical que FreeLing es incapaz de resolver. Por ejemplo, en la frase: *"Es preciso que uno de los tres muera."*, FreeLing detecta el verbo en subjuntivo —muera— como un sustantivo y lo etiqueta como N. Los sustantivos son candidatos a ser subsituidos en nuestros modelos. A partir de ahí, al construir una nueva frase con nuestros sistemas a partir de esta representación y usando el *query*="mundo", obtemos por ejemplo: *"Es necesario que uno de los tres* ***tierra***", que es incoherente.

El trabajo a futuro necesita la implementación de módulos para procesar los *queries* multitérmino del usuario. También se tiene contemplada la generación de frases retóricas en el dominio de discursos políticos, utilizando los modelos aquí propuestos u otros con un enfoque probabilístico (Charton & Torres-Moreno, 2010). Tenemos la hipótesis que una cierta atractividad del discurso político reside no tanto en el contenido mismo, sino en la estructura y en la manera de producir dichas frases (Cossu et al., 2014; Abascal-Mena et al., 2015).

Los modelos aquí presentados pueden ser enriquecidos a través de la integración de otros componentes, como características de una personalidad y/o las emociones (Wedemann & Carvalho, 2012; Wedemann & Plastino, 2016; Edalat, 2017; Siddiqui et al., 2018).

La introducción de la rima puede ser sumamente interesante cuando se produzcan varias frases para constituir un párrafo o una estrofa. El acoplamiento con un generador de rimas asonantes y consonantes (Medina-Urrea & Torres-Moreno, 2019) está previsto.

Finalmente, un protocolo de evaluación semiautomático (y a gran escala) está igualmente previsto.

Agradecimientos

Este trabajo está financiado por el Consejo Nacional de Ciencia y Tecnología (Conacyt, México), beca núm. 661101 y parcialmente por la Université d'Avignon, Laboratoire Informatique d'Avignon (LIA), programa de becas Agricol Perdiguier (France).

Los autores agradecen profundamente a los siete evaluadores anónimos que participaron en este trabajo; y a Carlos González por sus observaciones y sugerencias, asi como a los árbitros de la revista por sus comentarios pertinentes.

Referencias

Abascal-Mena, Rocío, Jean-Valère Cossu, Alejandro Molina-Villegas & Juan-Manuel Torres-Moreno. 2015. Anotación automática de datos acerca de la reputación de los políticos en redes sociales. *Research in Computing Science* 97. 81–99.

Agirrezabal, Manex, Bertol Arrieta, Aitzol Astigarraga & Mans Hulden. 2013. POS-tag based poetry generation with WordNet. En *14th European Workshop on Natural Language Generation*, 162–166.

Bengio, Yoshua, Aaron Courville & Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(8). 1798–1828. doi 10.1109/TPAMI.2013.50.

Boden, Margaret A. 2004. *The creative mind: Myths and mechanisms*. Abingdon: Routledge 2ª ed.

Bracewell, David B, Fuji Ren & Shingo Kuriowa. 2005. Multilingual single document keyword extraction for information retrieval. En *International Conference on Natural Language Processing and Knowledge Engineering*, 517–522. doi 10.1109/NLPKE.2005.1598792.

Charton, Eric & Juan-Manuel Torres-Moreno. 2010. Modélisation automatique de connecteurs logiques par analyse statistique du contexte. *Canadian Journal of Information and Library Science* 35(3). 287–306. 10.1353/ils.2011.0017.

Clark, Elizabeth, Yangfeng Ji & Noah A. Smith. 2018. Neural text generation in stories using entity representations as context. En *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2250–2260. 10.18653/v1/N18-1204.

Colton, Simon. 2012. *Automated theory formation in pure mathematics*. London: Springer. 10.1007/978-1-4471-0147-5.

Colton, Simon & Geraint A Wiggins. 2012. Computational creativity: The final frontier? En *20th European Conference on Artificial Intelligence*, 21–26.

Cossu, Jean-Valère, Rocío Abascal-Mena, Alejandro Molina-Villegas, Juan-Manuel Torres-Moreno & Eric SanJuan. 2014. Bilingual and cross domain politics analysis. *Research in Computing Science* 85. 9–19.

van Deemter, Kees, Mariët Theune & Emiel Krahmer. 2005. Real versus template-based natural language generation: A false opposition? *Computational Linguistics* 31(1). 15–24. 10.1162/0891201053630291.

Edalat, Abbas. 2017. Self-attachment: A holistic approach to computational psychiatry. En Péter Érdi, Basabdatta. Sen Bhattacharya & Amy L Cochran (eds.), *Computational Neurology and Psychiatry*, vol. 6, 273–314. Cham: Springer. 10.1007/978-3-319-49959-8_10.

Fu, Ruiji, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang & Ting Liu. 2014. Learning semantic hierarchies via word embeddings. En *52nd Annual Meeting of the Association for Computational Linguistics*, 1199–1209. 10.3115/v1/P14-1113.

Gervás, Pablo, Raquel Hervás & Carlos León. 2015. Generating plots for a given query using a case-base of narrative schemas. En *23rd International Conference on Case-Based Reasoning*, 103–112.

Gonçalo Oliveira, Hugo. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. En *10th International Conference on Natural Language Generation*, 11–20. 10.18653/v1/W17-3502.

Gonçalo Oliveira, Hugo. 2012. PoeTryMe: a versatile platform for poetry generation. En *1st International Workshop on Computational Creativity, Concept Invention and General Intelligence*, s.p.

Gonçalo Oliveira, Hugo & Amílcar Cardoso. 2015. Poetry generation with PoeTryMe. En *Computational Creativity Research: Towards Creative Machines*, 243–266. 10.2991/978-94-6239-085-0_12.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville & Yoshua Bengio. 2014. Generative adversarial nets. En Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, vol. 27, 2672–2680.

Huang, Eric H., Richard Socher, Christopher D. Manning & Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. En *50th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 873–882.

Iria, da Cunha, M. Teresa Cabré, Eric SanJuan, Gerardo Sierra, Juan-Manuel Torres-Moreno & Jorge Vivaldi. 2011. Automatic specialized vs. non-specialized sentence differentiation. En *12th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, 266–276. 10.1007/978-3-642-19437-5_22.

Ke, Wang & Wan Xiaojun. 2018. SentiGAN: Generating sentimental texts via mixture adversarial networks. En *27th International Joint Conference on Artificial Intelligence (IJCAI)*, 4446–4452. 10.24963/ijcai.2018/618.

Lebret, Rémi, David Grangier & Michael Auli. 2016. Neural text generation from structured data with application to the biography domain. En *Conference on Empirical Methods in Natural Language Processing*, 1203–1213. 10.18653/v1/D16-1128.

Manning, Christopher D. & Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. Cambridge: The MIT Press.

Martínez, Gerardo Sierra. 2018. *Introducción a los corpus lingüísticos*. Mexico: Instituto de Ingeniería, UNAM.

McRoy, Susan, Songsak Channarukul & Syed Ali. 2003. An augmented template-based approach to text realization. *Natural Language Engineering* 9(4). 381–420. 10.1017/S1351324903003188.

Medina-Urrea, Alfonso & Juan-Manuel Torres-Moreno. 2019. RIMAX: ranking semantic rhymes by calculating definition similarity. *arXiv* CoRR abs/1912.09558.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado & Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. En *26th International Conference on Neural Information Processing Systems*, 3111–3119.

Mikolov, Tomas, Wen-tau Yih & Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. En *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–751.

Mikolov, Tomas & Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. En *IEEE Spoken Language Technology Workshop (SLT)*, 234–239. 10.1109/SLT.2012.6424228.

Molins, Paul & Guy Lapalme. 2015. JSrealB: A bilingual text realizer for web programming. En *15th European Workshop on Natural Language Generation (ENLG)*, 109–111. 10.18653/v1/W15-4719.

Montfort, Nick. 2008b. Through the park. https://nickm.com/poems/through_the_park.html.

Montfort, Nick. 2008c. The two. https://nickm.com/poems.

Montfort, Nick. 2009. Taroko gorge. https://nickm.com/poems/taroko_gorge.html.

Padró, Lluís. 2012. Analizadores multilingües en freeling. *Linguamática* 3(2). 13–20.

Pérez y Pérez, Rafael. 2015. *Creatividad computacional*. México: Larousse, Grupo Editorial Patria.

Riedl, Mark O. & R. Michael Young. 2006. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing* 24(3). 303–323. 10.1007/BF03037337.

Sharples, Mike. 1996. *How we write: Writing as creative design*. London: Routledge.

Siddiqui, Maheen, Roseli S. Wedemann & Henrik Jeldtoft Jensen. 2018. 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. 10.1016/j.physa.2017.08.011.

Sridhara, Giriprasad, Emily Hill, Divya Muppaneni, Lori Pollock & K. Vijay-Shanker. 2010. Towards automatically generating summary comments for java methods. En *IEEE/ACM International Conference on Automated Software Engineering*, 43–52. 10.1145/1858996.1859006.

Szymanski, Grzegorz & Zygmunt Ciota. 2002. Hidden markov models suitable for text generation. En *International Conference on Signal, Speech and Image Processing*, 3081–3084.

Torres-Moreno, Juan-Manuel. 2012. Beyond stemming and lemmatization: Ultra-stemming to improve automatic text summarization. *arXiv* abs/1209.3126.

Torres-Moreno, Juan-Manuel (ed.). 2014. *Automatic text summarization*. London: ISTE, Wiley.

Wedemann, Roseli S. & Luiz Alfredo Vidal de Carvalho. 2012. Some things psychopathologies can tell us about consciousness. En *International Conference on Artificial Neural Networks (ICANN)*, vol. 7552, 379–386. 10.1007/978-3-642-33269-2_48.

Wedemann, Roseli S. & Angel Ricardo Plastino. 2016. Física estadística, redes neuronales y freud. *Revista Núcleos* 3. 4–10.

Welleck, Sean, Kiante Brantley, Hal Daumé & Kyunghyun Cho. 2019. Non-monotonic sequential text generation. En *36th International Conference on Machine Learning*, 11656–11676.

Zhang, Xingxing & Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. En *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 670–680. 10.3115/v1/D14-1074.

A. Frases evaluadas

En este anexo presentamos las 45 (15 frases × 3 sistemas) frases generadas por nuestros modelos, que fueron evaluadas manualmente (ver sección 5).

Modelo 1

1. $f(\texttt{GUERRA}, 10)$ = *La simpatía es el aprecio que ha olvidado la ternura.*
2. $f(\texttt{GUERRA}, 13)$ = *Toda batalla en rebelión es la guerra contra el ejército en el combate.*
3. $f(\texttt{GUERRA}, 12)$ = *El ejército conquista mediante el enemigo. La batalla es la guerra desde.*
4. $f(\texttt{GUERRA}, 13)$ = *El enemigo salvo la batalla en el terrorismo mediante el ejército conquista contra.*
5. $f(\texttt{GUERRA}, 11)$ = *Su atómica lucha. la guerra desde el combate. la derrota es.*
6. $f(\texttt{SOL}, 12)$ = *La luna salvo la lluvia sobre el ocaso hacia el cielo brilla.*
7. $f(\texttt{SOL}, 13)$ = *Cuántos naveguen salvo iluminar para el cielo hacia la aurora es la luna.*
8. $f(\texttt{SOL}, 13)$ = *Nuestro cielo es verdaderamente el que luna es la lluvia bajo la aurora.*
9. $f(\texttt{SOL}, 9)$ = *Cuántos perezcas durante amanecer el ocaso bajo la luna.*
10. $f(\texttt{SOL}, 6)$ = *El resplandor deshoja bajo la aurora.*
11. $f(\texttt{AMOR}, 13)$ = *Toda amistad contra compasión por la ternura es la pasión en el afecto.*
12. $f(\texttt{AMOR}, 13)$ = *Todos durante la ternura con el afecto hacia la compasión por la virtud.*
13. $f(\texttt{AMOR}, 12)$ = *El cariño con el afecto hacia el amado aborrece salvo en sentimiento.*
14. $f(\texttt{AMOR}, 11)$ = *La ternura envidia entre el cariño. El amado es demasiada compasión.*
15. $f(\texttt{AMOR}, 11)$ = *Cuánto bien deseo sin la amistad. El afecto es otra ternura.*

Modelo 2

1. $f(\texttt{GUERRA}, 9)$ = *El incivil comportamiento para la magnificencia es la dicha.*
2. $f(\texttt{GUERRA}, 9)$ = *Mi anciana: tú felicidad no la alumbra ninguna autoridad.*
3. $f(\texttt{GUERRA}, 9)$ = *No hay hipocresía más impopular que la historia simulada.*
4. $f(\texttt{GUERRA}, 10)$ = *La cultura es la religión de dogmatizar los bienes caducos.*
5. $f(\texttt{GUERRA}, 10)$ = *De el temperamento a la entereza hay una velocidad terrible.*
6. $f(\texttt{SOL}, 9)$ = *El color que reanima más es una picardía suprema.*
7. $f(\texttt{SOL}, 10)$ = *La paz es el suelo artificial de la luz moderna.*
8. $f(\texttt{SOL}, 10)$ = *En el vocabulario está el bosque mixto de una política.*
9. $f(\texttt{SOL}, 10)$ = *El rocío exhala el bosque después de haber lo fatigado.*
10. $f(\texttt{SOL}, 11)$ = *Brilla que contener siempre. Nunca se es dominado de el todo.*
11. $f(\texttt{AMOR}, 9)$ = *Estorbas una amada calle de un amor de fantasías.*
12. $f(\texttt{AMOR}, 9)$ = *Jamás hubo una conquista nueva o una amistad extraña.*
13. $f(\texttt{AMOR}, 11)$ = *Dios dejó la desesperacion para trabajar la y no para desilusionarla.*
14. $f(\texttt{AMOR}, 11)$ = *Abultar se en cualquier mentira, es conveniente que no porfiar nada.*
15. $f(\texttt{AMOR}, 10)$ = *Por culpa, el anhelo no suprime siempre con el deseo.*

Modelo 3

1. $f(\texttt{GUERRA}, 9)$ = *Existe demasiada innovacion en torno a muy pocos sucesos.*
2. $f(\texttt{GUERRA}, 9)$ = *En la pelea todo debe motivo, menos la retirada.*
3. $f(\texttt{GUERRA}, 10)$ = *La nueva pelea se combate cuando se abandona la civilización.*
4. $f(\texttt{GUERRA}, 10)$ = *La codicia, siempre adversa, es terrible engendrada contra un desgraciado.*
5. $f(\texttt{GUERRA}, 10)$ = *La retirada es el vapor remediable de el lucha ilimitada.*
6. $f(\texttt{SOL}, 9)$ = *Si tus dulces fueran amanecer, mis ojos marchitas fueran.*
7. $f(\texttt{SOL}, 11)$ = *Con rapidez, los monógamos impedimentos buscan para iluminar nos la luz.*
8. $f(\texttt{SOL}, 10)$ = *Incluso los luceros ingratos son comilones, y por tanto antiguos.*
9. $f(\texttt{SOL}, 9)$ = *El ocaso es una extraña frente de la inmortalidad.*
10. $f(\texttt{SOL}, 9)$ = *La aurora es el amanecer que ha olvidado la calma.*
11. $f(\texttt{AMOR}, 10)$ = *En el aprecio está el cariño forzoso de una simpatía.*
12. $f(\texttt{AMOR}, 10)$ = *Los cariños no conocen de nada a un respeto loco.*
13. $f(\texttt{AMOR}, 10)$ = *No está la simpatía en las bondades de la envidia.*
14. $f(\texttt{AMOR}, 9)$ = *Si el respeto es felicidad, que oculten los cariños.*
15. $f(\texttt{AMOR}, 10)$ = *Acostumbramos de lamentar aquello que se ha enseñado a comprender.*