

Estudio de hiperparámetros de modelos neuronales en la generación de frases literarias

Luis-Gil Moreno-Jiménez¹, Juan-Manuel Torres-Moreno^{1,4},
Carlos-Emiliano González-Gallardo², Roseli S. Wedemann³

¹ Avignon Université,
Laboratoire Informatique d'Avignon,
France

² Sorbonne Université,
Laboratoire STIH,
France

³ Universidade do Estado do Rio de Janeiro,
Brazil

⁴ Polytechnique Montréal,
Canada

`luis-gil.moreno-jimenez@alumni.univ-avignon.fr,`
`juan-manuel.torres@univ-avignon.fr,`
`carlos-emiliano.gonzalez-gallardo@sorbonne-universite.fr,`
`roseli@ime.uerj.br`

Resumen. Los algoritmos de Redes Neuronales son ampliamente utilizados en técnicas de Inteligencia Artificial (IA) para la resolución de problemas en diferentes campos de la ciencia que no pueden ser resueltos por la computación simbólica tradicional. En este trabajo, presentamos un conjunto de experimentos que involucran la implementación de varios modelos basados en word2vec, una representación de palabras basada en la semántica distribucional. Nuestros experimentos se desarrollaron con el propósito de establecer un conjunto de métricas replicables para el entrenamiento de un modelo word2vec capaz de ser integrado a un modelo para generar oraciones literarias coherentes y con un contexto bien definido. En este trabajo, presentamos resultados alentadores del análisis de oraciones generadas con nuestros modelos mediante evaluaciones manuales.

Palabras clave: Redes neuronales artificiales, procesamiento del lenguaje natural, generación de textos literarios, word2vec.

Study of Hyperparameters of Neural Models in the Generation of Literary Sentences

Abstract. Neural Network algorithms are widely used in Artificial Intelligence (AI) techniques for solving problems in different fields of science that cannot be solved by traditional symbolic computation. In this paper, we present a set of

experiments involving the implementation of various models based on word2vec, a word representation based on distributional semantics. Our experiments were developed with the purpose of establishing a set of replicable metrics for training a word2vec model capable of being integrated into a model to generate coherent literary sentences with a well-defined context. In this paper, we present encouraging results from the analysis of sentences generated with our models using manual evaluations.

Keywords: Artificial neural networks, natural language processing, literary text generation, word2vec.

1. Introducción

Los modelos basados en Redes Neuronales (NN por sus siglas en inglés) han sido ampliamente usados en diversos campos de estudio, dada su eficiencia y efectividad en métodos predictivos, para el análisis de datos y tareas de tipo cognitivas. Estos modelos son idóneos cuando se requiere procesar y analizar una cantidad importante de datos. En Procesamiento de Lenguaje Natural (PLN), las NN son comúnmente usadas en tareas de clasificación de textos, recuperación de información, traducción automática, entre otras [6, 9, 17].

En el campo de la literatura, estos modelos neuronales son usados para crear estructuras complejas del lenguaje para la generación de texto literario [18, 20] en un área denominada Creatividad Computacional [3]. En Creatividad Computacional las NN son entrenadas con cantidades importantes de datos a partir de ciertos artefactos artísticos como: la pintura, la música, novelas, cuentos, etc. La intención es detectar patrones para imitar el Proceso Creativo⁵ que permite la creación de cada obra artística.

En este trabajo analizamos cómo los diferentes hiperparámetros (dimensión, ventana contextual, ocurrencias, etc.) pueden ser indicadores interesantes para el entrenamiento de modelos basados en representaciones vectoriales Word2vec [8] utilizados en la generación de frases literarias. Con el fin de estudiar el efecto de los hiperparámetros, integramos nuestros experimentos al modelo propuesto en [11], donde se realiza un estudio semántico-contextual para la generación de texto literario.

Word2vec produce un vector numérico (embeddings) para cada palabra dentro de un vocabulario de aprendizaje. Estos vectores son calculados bajo un parametraje específico, de acuerdo a las características del corpus y los objetivos pretendidos. A partir de los embeddings es posible calcular la distancia entre estos, y determinar cuáles están más próximos entre sí, estableciendo un primer análisis de tipo semántico.

Este estudio exploratorio basado en Word2vec puede servir para determinar el parametraje que mejor se ajusta a un corpus literario, la intención es obtener una aproximación semántica adecuada para la generación de frases coherentes. Para ello, buscamos que los embeddings produzcan una semántica de tipo literaria. Por ejemplo,

⁵ Término explicado ampliamente en [1], donde establecen algunas directrices para estudiar las habilidades cognitivas del hombre desde un enfoque computacional.

dada la palabra “azul” se esperaría que las palabras cercanas fueran “mar” o “cielo”, y no “verde” o “rojo”.

En la sección 2, presentamos el resumen de algunos trabajos con el mismo enfoque sobre implementaciones de modelos Word2vec. El corpus usado para el entrenamiento de nuestros modelos es descrito en la sección 3. Posteriormente, en la sección 4 detallamos los distintos parámetros usados para el entrenamiento, así como algunas gráficas con resultados.

En la sección 5 mostramos algunas frases generadas por los modelos que consideramos mejor entrenados, de acuerdo a nuestro criterio de dispersión. Los resultados de la evaluación manual efectuada sobre las frases generadas son presentados en la sección 6. Finalmente, en la sección 7 se presentan las conclusiones sobre los resultados.

2. Estado del arte

Los modelos de NN han sido abordados con mayor determinación en los últimos años por investigadores del área de Creatividad Computacional, sobretodo para la generación de texto literario. A continuación presentamos algunos trabajos relacionados.

2.1. Generación de texto no literario

Lebret et al. [6] proponen un algoritmo para generar frases biográficas, que es entrenado con un conjunto de datos biográficos extraídos de Wikipedia⁶. Otro modelo para la generación automática de texto se presenta en [16]. En este trabajo se propone la generación de descripciones para fragmentos de código Java. En el modelo se consideran elementos como: nombre de los constructores, de las llamadas, de las instancias, etc., para generar una descripción apropiada.

Otras implementaciones basadas en Word2vec para la generación automática de textos son propuestas en [4], donde Kharazmi et al. analizaron los embeddings propuestos por su modelo y los combinaron con una serie de n-gramas, con la intención de calcular la coherencia de un documento. Se tiene también el trabajo de Kiddon et al. [5], presentan un modelo de Word2vec para conservar la coherencia en el texto generado.

2.2. Generación de texto literario

Zhang et al. [20] proponen un modelo basado en Redes Neuronales Recurrentes para la generación de poesía en chino, el aprendizaje se basa en el reconocimiento de estructuras lingüísticas. Las estructuras analizadas se convierten en vectores que luego se usan para calcular la probabilidad de aparición de la siguiente estructura.

Otra propuesta basada en NN para poesía es presentada en [12], este modelo recibe una palabra como entrada y construye el texto con elementos relacionados a ésta.

Tabla 1. Características de **MegaLite** y **cGoethe** corpus (*K*: mil, *M*: millón).

	MegaLite	cGoethe
Frases	9.2 M	19 519
Tokens	99.7 M	340 K
Media por frase	11	17
Letras	604 M	2 M
Media por frase	65	103

Existen investigaciones, como el proyecto MEXICA [13], donde se ha logrado generar textos literarios con una longitud considerable, sobrepasando la barrera a nivel de frases.

MEXICA genera cuentos sobre la época pre-colombina de la mitología Azteca empleando el proceso denominado engagement-reflection propuesto en [15]. Otro algoritmo basado en NN para el análisis contextual fue propuesto en [2] para la generación de historias de ficción.

Finalmente, en [11] se propone un modelo para la generación de frases literarias, donde los embeddings son procesados bajo una interpretación geométrica para seleccionar las palabras que mejor obedecen a un contexto definido y formar frases coherentes y semánticas.

3. Corpus

En este trabajo empleamos dos corpus con textos literarios en Español (algunos traducidos de otros idiomas). El primer corpus es MegaLite [10], que contiene aproximadamente 5 000 documentos (en su mayoría libros), los cuales corresponden a los géneros: cuentos, poesía, teatro y ensayos. Por la cantidad considerable de textos contenidos, el corpus MegaLite fue utilizado para los distintos entrenamientos de las instancias basadas en la representación vectorial Word2vec.

El segundo corpus, cGoethe, está constituido por las obras principales de Johann Wolfgang von Goethe. La utilidad de este corpus es identificar y replicar las asociaciones semánticas que el autor emplea al momento de redactar sus obras. El corpus cGoethe fue usado para la fase de generación textual que se describe en la sección 5, dedicada a experimentos. Para construir ambos corpus, se convirtió cada documento al formato de codificación de caracteres utf8.

Además, se segmentaron los textos de cada documento en una frase por línea empleando la biblioteca NLTK⁷ de Python. Posteriormente se seleccionó la novela “Los lamentos del joven Werther” [19] de Goethe, esto por la fuerte carga emocional y ciertos aspectos psicológicos observados en los personajes principales. De esta novela, se eligieron manualmente las frases que se consideraron como “literarias”, es decir, se determinó que estas frases contenían un vocabulario estético, además de tener ciertas figuras literarias como la rima, la anáfora, la paráfrasis y otras.

Con estas frases se construyó el subconjunto cWerther utilizado en la fase de generación textual descrito en la sección 5. En la tabla 1 se muestra información

⁶ <https://www.wikipedia.org>

⁷ <https://www.nltk.org/api/nltk.tokenize.html>

Tabla 2. Características del subconjunto **cWerther**.

	cWerther	Media por frase
Frases	134	—
Tokens	1 635	12
Letras	9 321	69

Tabla 3. Valores de parámetros para los modelos Word2vec.

Parámetro	Valores
Iteraciones (i)	1, 5, 7, 10
Conteo mínimo (m)	3, 5
Tamaño del vector (s)	60, 100
Tamaño de la ventana (w)	5, 6, 7

estadística sobre los corpus MegaLite y cGoethe. El corpus cWerther es descrito en la tabla 2.

4. Representación vectorial de palabras

Uno de los intereses principales de este estudio es analizar el impacto de las representaciones vectoriales de palabras en la generación de frases literarias. Por esto entrenamos 48 diferentes modelos de Word2vec utilizando el corpus MegaLite. La tabla 3 muestra los diferentes parámetros y valores utilizados para entrenar los modelos. “Iteraciones”(i) hace referencia al número de ciclos de entrenamiento completados con el corpus MegaLite.

“Conteo mínimo”(m) indica el número mínimo de veces que una palabra debe aparecer en el corpus para ser incluida en el vocabulario del modelo. “Tamaño del vector”(s) especifica la dimensión de los vectores de palabras. Por último, “Tamaño de la ventana”(w) representa la cantidad de palabras adyacentes (contexto) que son relacionadas a una palabra estudiada en una oración durante la fase de entrenamiento.

Entrenamos todos los modelos siguiendo el enfoque de skip-gram [8] con un muestreo negativo de cinco palabras y un umbral de submuestreo igual a 0.001. El tamaño del vocabulario para los modelos con $m = 3$ consta de 295 838 palabras, mientras que para los modelos con $m = 5$ el vocabulario es igual a 222 095 palabras.

Para analizar la relación entre los 48 diferentes modelos de Word2vec, realizamos una evaluación basada en los trabajos de Pierrejean y Tanguy [14] donde se comparan pares de modelos. Este método estima la tasa de variación global entre un par de modelos M_a y M_b a partir de un vocabulario Q .

En primera instancia, definimos $\text{sim}_{M_a}^N(q)$ como el conjunto de las N palabras en M_a más similares (en términos de su similitud coseno) a una palabra objetivo q ; de igual forma definimos $\text{sim}_{M_b}^N(q)$. La tasa de variación de una palabra $q \in Q$ depende entonces del número de palabras en la intersección de los conjuntos $\text{sim}_{M_a}^N(q)$

y $\text{sim}_{M_b}^N(q)$, que está definida como:

$$\text{var}_{M_a, M_b}^N(q) = 1 - \frac{|\text{sim}_{M_a}^N(q) \cap \text{sim}_{M_b}^N(q)|}{N}, \quad (1)$$

donde $|\text{sim}_{M_a}^N(q) \cap \text{sim}_{M_b}^N(q)|$ corresponde al número de palabras (la cardinalidad) en la intersección. Finalmente, la tasa de variación global Gvar_{M_a, M_b}^N , entre M_a y M_b sobre Q , con cardinalidad $|Q|$, es definida como:

$$\text{Gvar}_{M_a, M_b}^N = \frac{\sum_{q \in Q} \text{var}_{M_a, M_b}^N(q)}{|Q|}. \quad (2)$$

Pierrejean y Tanguy [14] condujeron un estudio para determinar el mejor valor de N . Encontraron que para $N = 25$, la correlación media entre las tasas de variación, obtenida a partir de un conjunto de diferentes valores de N era maximizada. Basándose en este resultado, fijamos el valor de N a 25. Para crear el vocabulario Q , seleccionamos de **MegaLite** sólo aquellas palabras con un mínimo de 100 ocurrencias pertenecientes a alguna de las siguientes categorías gramaticales: adjetivo, adverbio, sustantivo y verbo.

Después de este proceso de filtrado, el número total de palabras contenidas en Q es 42 681. Por cada una de las siguientes palabra objetivo, obtuvimos las 25 palabras más similares a partir de la similitud coseno de sus representaciones vectoriales: “amor”, “odio”, “tristeza”, “alegría”, “miedo”, “mujer”, “hombre” y “azul”. La figura 1 muestra los diagramas de caja resultantes de los 48 modelos ordenados por similitud coseno media. En esta figura se observa que aquellos modelos con una sola iteración y tamaño del vector más pequeño obtienen resultados más altos.

En contraste, los modelos con varias iteraciones y tamaño del vector más grande obtienen menores resultados. También es posible observar que, en general, los modelos con similitudes promedio más grandes presentan una menor dispersión al compararlos con aquellos modelos con similitudes promedio menores. Este comportamiento puede ser mejor observado en la figura 2, en donde se muestra, por cada modelo, la desviación estándar de las similitudes coseno agrupadas por palabra objetivo.

Al comparar ambas figuras, se puede ver que para los modelos con una sola iteración, los resultados de similitud son más altos y las desviaciones estándar más pequeñas. En contraste, los modelos con varias iteraciones y tamaño del vector más grande presentan resultados de similitud más bajos y desviaciones estándar más altas. De hecho, existe una fuerte correlación lineal negativa igual a -0.96 entre las similitudes coseno agrupadas por palabra objetivo y sus correspondientes desviaciones estándar. Con estos resultados, seleccionamos los mejores y peores modelos en términos de dispersión de similitud de coseno, sobre el conjunto de palabras objetivo.

5. Generación textual

En esta sección describimos brevemente el modelo presentado en [11], ilustrado en la figura 3, para la generación automática de frases literarias. El modelo consiste en dos etapas, siendo la segunda etapa donde integramos nuestra propuesta de modelo de

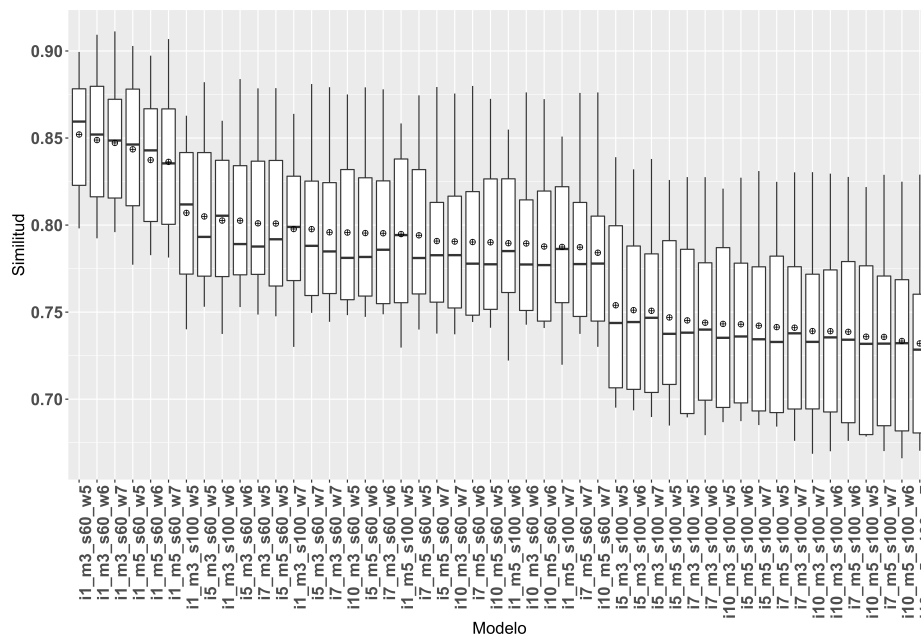


Fig. 1. Similitud coseno por cada modelo Word2vec.

representación vectorial Word2vec. El proceso inicia con una palabra (query) provista por el usuario. Esta palabra define el contexto para la generación de la nueva frase. La frase generada se construye reemplazando el vocablo de una frase original escrita por un humano, ésta es extraída del conjunto cWerther. Esta frase debe conservar el estilo y el contenido emocional que el autor expresa en la frase original.

Durante la primera etapa, el conjunto cWerther es procesado con FreeLing⁸, un analizador morfosintáctico. Por cada frase original se detectan y reemplazan las palabras léxicas⁹ por sus etiquetas gramaticales (POS) correspondientes. El resto de las palabras son conservadas. Como resultado una Estructura Gramatical semi-Vacía (EGV) es generada, esta EGV se compone entonces de etiquetas POS y palabras funcionales (artículos, conectores, adverbios, etc.).

Este método, basado en estructuras lingüísticas fijas, se conoce como Canned Text y ha sido empleado en trabajos donde se proponen modelos para la generación rápida de diálogos y frases cortas [18, 7]. En la segunda etapa, las etiquetas POS de la EGV se reemplazan con un vocabulario semánticamente ad-hoc.

Es decir, los sustantivos deben acercarse más al nuevo contexto, definido por el usuario, mientras que los verbos y adjetivos deben estar más próximos al contexto de la frase original. El vocabulario utilizado para el proceso de reemplazo se obtiene de los modelos Word2vec descritos en la sección 4. En esta etapa, el modelo generativo emplea tres elementos importantes:

- **PO**: La palabra que correspondía a la frase original.

⁸ FreeLing se encuentra disponible en <http://PLN.lsi.upc.edu/freeling>

⁹ Sustantivos, verbos principales y adjetivos.

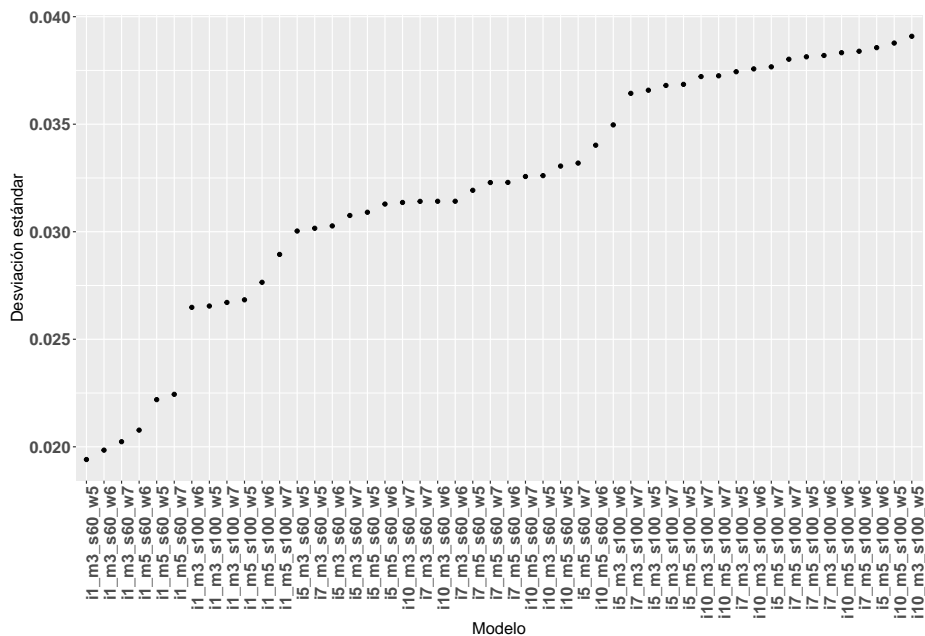


Fig. 2. Desviación estándar por cada modelo Word2vec.

- **VG**: El vocabulario extraído del corpus cGoethe.
- **Q**: El nuevo contexto.

A través de operaciones matemáticas y cálculos de distancias entre vectores, podemos utilizar los modelos Word2vec para elegir del vocabulario VG la palabra más próxima a PO, si se trata de un verbo o adjetivo; si se trata de un sustantivo, se elige la palabra más cercana a Q. Más detalles del modelo generativo pueden consultarse en [11].

A continuación mostramos algunas frases generadas en la fase experimental de este proyecto. Las frases fueron generadas usando los cuatro modelos de Word2vec considerados como relevantes según nuestro criterio de dispersión (ver sección 4). Las instancias elegidas son: i1_m3_s60_w5 ($i=1, m=3, s=60, w=5$), i1_m3_s60_w6 ($i=1, m=3, s=60, w=6$), i10_m5_s100_w5 ($i=10, m=5, s=100, w=5$) y i10_m3_s100_w5 ($i=10, m=3, s=100, w=5$).

Por motivos prácticos, de aquí en adelante llamaremos a estas instancias: m3s60w5, m3s60w6, m5s100 w5 y m3s100w5, respectivamente. En los ejemplos a continuación se muestran las frases originales con las palabras reemplazables en **negrita**, seguidas por las nuevas frases guiadas por (Q).

- Instancia m3s60w5.
 Frase original: Pero en sus [cercanías] la [naturaleza] [brilla].
 – $f(\text{AMOR}) =$ Pero en sus penas forma la gloria.
 – $f(\text{ODIO}) =$ Pero en sus mejillas forma la alegría.

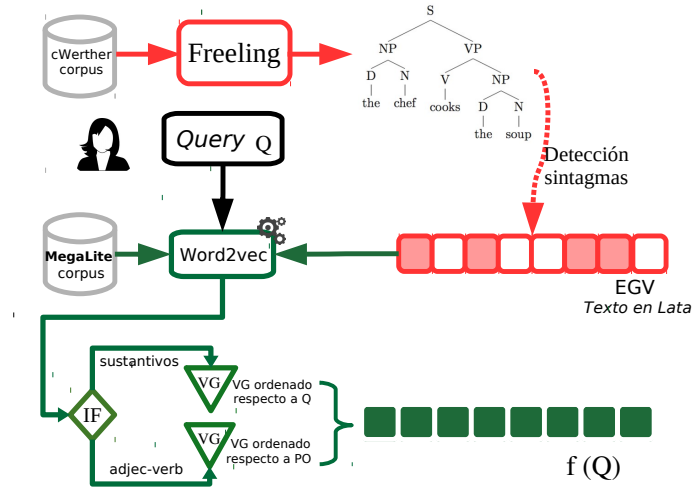


Fig. 3. Estructura general del modelo generativo de frases literarias.

- $f(\text{MUJER}) = \text{Pero en sus hermanas desaparece la mujer.}$
- Instancia m3s60w6.
 Frase original: El [buen] o [mal] [humor] no [obedece] a nuestra [voluntad].
 - $f(\text{AMOR}) = \text{El seguro o loco amor no cuenta a nuestra alma.}$
 - $f(\text{ODIO}) = \text{El necesario o loco amor no considera a nuestra desesperación.}$
 - $f(\text{MUJER}) = \text{El único o buen padre no considera a nuestra amiga.}$
- Instancia m5s100w5.
 Frase original: El [llanto] se [agolpó] en mis [ojos].
 - $f(\text{AMOR}) = \text{El amor se agolpó en mis deseos.}$
 - $f(\text{ODIO}) = \text{El miedo se agolpó en mis sentimientos.}$
 - $f(\text{MUJER}) = \text{El niño se agolpó en mis padres.}$
- Instancia m3s100w5.
 Frase original: ¡Qué [gracia], qué [agilidad] en sus [movimientos]!
 - $f(\text{AMOR}) = \text{¡Qué pasión, qué felicidad en sus deseos!}$
 - $f(\text{ODIO}) = \text{¡Qué desesperación, qué angustia en sus sentimientos!}$
 - $f(\text{MUJER}) = \text{¡Qué mujer, qué criada en sus hijos!}$

Los ejemplos anteriores fueron generados a partir de frases originales tomadas aleatoriamente del corpus cWerther. Se observa que, en algunos casos, algunas palabras no son reemplazadas, esto ocurre cuando una palabra no es encontrada dentro del modelo Word2vec.

6. Evaluación

Se efectuó una evaluación de los modelos Word2vec entrenados en los experimentos de la sección 5. La evaluación fue manual y para ello se consideró un número asequible de frases que el evaluador debía analizar. Las frases se generaron utilizando

Tabla 4. Distribución de las frases evaluadas.

Descripción	Primer Conjunto	Segundo Conjunto	Tercer Conjunto	Total
Número de frases	16	16	8	40
Modelos Word2vec	m3s60w5	m3s100w5	Frase original	—
Contexto (queries)	AMOR, ODIO, MUJER		N/A	—

dos modelos Word2vec, el de mayor y menor valor de dispersión, m3s100w5 y m3s60w5 respectivamente.

Seis personas fueron elegidas como evaluadores, estas son hablantes nativos del español, con grados universitarios y cuentan con cierta experiencia literaria, sea como lectores frecuentes o escritores amateurs. Se les pidió evaluar 40 frases (todas en español): 16 del modelo m3s60w5, 16 del modelo m3s100w5 y 8 frases humanas tomadas directamente de la obra traducida: Los lamentos del joven Werther. Los criterios a evaluar fueron:

- **Gramática:** Conjugaciones correctas y adecuada relación entre género y número.
- **Coherencia:** Las palabras en la frase deben situarse dentro del mismo contexto.
- **Percepción literaria:** Indicar si se consideraban las frases como literarias o no.
- **Percepción emocional:** Indicar la presencia de alguna de las siguientes emociones: Miedo, Tristeza, Esperanza, Amor y Felicidad.

Las frases de ambas instancias fueron mezcladas junto con las frases humanas sin ninguna preferencia antes de ser entregadas a los evaluadores. La distribución de las frases evaluadas se muestra en la tabla 4. Para los criterios: Gramática y Coherencia, se calculó un valor de precisión, definido como el número de palabras correctas entre el número total de palabras reemplazables (léxicas), esto para cada frase. Luego se tomó el valor promedio de precisión calculado entre todas las frases para obtener un valor representativo para cada criterio, los resultados se muestran en la figura 4a.

Se puede observar que el modelo m3s100w5 (en azul) obtuvo mejores resultados para Coherencia con un 83.77 %, mientras que para Gramática, su desempeño fue menor en comparación con el modelo m3s60w5 (en rojo), sin embargo, el resultado es alentador con una precisión de 88.75 %. Para el criterio de Percepción literaria, el modelo m3s100w5 también obtuvo el mejor resultado, con un 64.7 % de las frases percibidas como literarias. Los resultados obtenidos muestran una mejora a los presentados en [11] donde si bien, para Gramática, las evaluaciones fueron aceptables; para Coherencia, las frases no fueron bien percibidas.

Se puede apreciar que el modelo m3s60w5 arroja embeddings más cercanos a un query determinado, esto genera frases más coherentes y permite manipular el nuevo contexto de la frase según se desee. Siguiendo nuestra estrategia (ver sección 5) de aproximar los sustantivos al nuevo contexto, dejando los verbos y adjetivos próximos al contexto original, conseguimos generar una frase con un contexto nuevo, preservando la carga emocional original. Esto puede ser observado en la figura 4b, donde se aprecian los resultados del modelo m3s100w5, con frases fuertemente relacionadas a

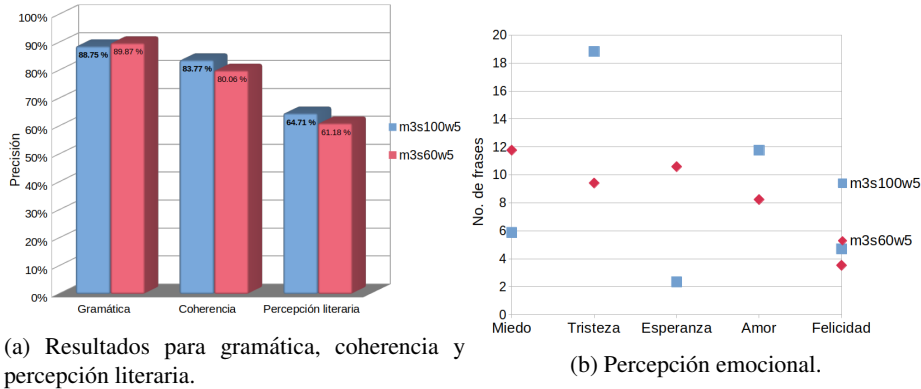


Fig. 4. Resultados de la evaluación manual en frases generadas.

las emociones “amor” y “tristeza”, siendo estas las principales emociones percibidas en la obra “Los lamentos del joven Werther”.

Para el modelo m3s60w5, la percepción parece no seguir un patrón regular, generando frases con una carga emocional aleatoria.

7. Conclusiones y trabajo futuro

En este trabajo hemos presentado resultados de experimentos con distintas implementaciones basadas en Word2vec para la generación de texto literario. Los resultados sugieren que la dispersión calculada entre los embeddings puede ser un indicador importante de la asociación semántica dentro de un espacio vectorial. El hiperparámetro discriminatorio que impacta en la dispersión de los modelos es el número de dimensiones¹⁰ de la representación. De acuerdo a la literatura, a mayor tamaño del corpus, es necesaria una mayor dimensión para obtener una mejor exactitud en tareas de asociación de palabras [8]. Lo cual también es una intuición razonable. Nuestros resultados con los evaluadores humanos confirman esta intuición.

Las instancias seleccionadas para nuestros experimentos demostraron un comportamiento adecuado, generando frases más coherentes que las reportadas en [11]. La instancia con menor dispersión generó frases con una carga emocional y una coherencia aceptable. Esto significa que bajo un entrenamiento idóneo, un modelo Word2vec puede ayudar a la generación de frases con un contexto bien definido. En el futuro pretendemos extender los experimentos utilizando corpus periodísticos, técnicos o científicos. Por otra parte, pensamos también estudiar la inclusión de la rima combinada con los modelos Word2vec presentados para generar parejas de frases literarias.

Agradecimientos. Este trabajo fue parcialmente financiado por el Consejo Nacional de Ciencia y Tecnología (CONACYT, Mexico), beca número 661101 y por la Fédération de Recherche AGORANTIC – Avignon Université, (France).

¹⁰ Este hiperparámetro es considerado al configurar el entrenamiento de un modelo Word2vec.

Referencias

1. Boden, M. A.: *The creative mind: Myths and Mechanisms*. Routledge (2004)
2. Clark, E., Ji, Y., Smith, N. A.: Neural text generation in stories using entity representations as context. *North American Chapter of the Association for Computational Linguistics*, vol. 1, pp. 2250–2260 (2018)
3. Colton, S., Wiggins, G. A.: Computational creativity: The final frontier? In: *20th European Conference on Artificial Intelligence*. Association for Computational Linguistics, pp. 21–26 (2012)
4. Kharazmi, M. A., Kharazmi, M. Z.: Text coherence new method using word2vec sentence vectors and most likely n-grams. In: *3rd Iranian Conference on Intelligent Systems and Signal Processing*, pp. 105–109 (2017) doi: 10.1109/ICSPIS.2017.8311598
5. Kiddon, C., Zettlemoyer, L., Choi, Y.: Globally coherent text generation with neural checklist models. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 329–339 (2016)
6. Lebret, R., Grangier, D., Auli, M.: Neural text generation from structured data with application to the biography domain. *Empirical Methods in Natural Language Processing*, pp. 1203–1213 (2016) doi: 10.48550/ARXIV.1603.07771
7. McRoy, S., Channarukul, S., Ali, S.: An augmented template-based approach to text realization. *Natural Language Engineering*, vol. 9, pp. 381–420 (2003)
8. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: *1st International Conference on Learning Representations*, International Conference on Learning (2013) doi: 10.48550/ARXIV.1301.3781
9. Molins, P., Lapalme, G.: JSrealB: A bilingual text realizer for web programming. In: *15th European Workshop on Natural Language Generation*, Association for Computational Linguistics, pp. 109–111 (2015)
10. Moreno, L. G., Torres, J. M.: Megalite: A new spanish literature corpus for nlp tasks. In: *8th International Conference on Artificial Intelligence and Applications* (2021)
11. Moreno, L. G., Torres, J. M., Wedemann, R. S.: Literary natural language generation with psychological traits. In: *Natural Language Processing and Information Systems*, Springer International Publishing, pp. 193–204 (2020)
12. Oliveira, H. G., Cardoso, A.: Poetry generation with poetryme. *Computational Creativity Research: Towards Creative Machines Atlantis Thinking Machines*, vol. 7 (2015)
13. Pérez y Pérez, R.: *Creatividad Computacional*. Editorial Patria (2015)
14. Pierrejean, B., Tanguy, L.: Reproducibility of word embeddings: Identifying stable and unstable zones in the semantic space. *Traitement Automatique des Langues Naturelles*, vol. 1 (2018)
15. Sharples, M.: *How We Write: Writing as creative design*. Routledge (1996)
16. Sridhara, G., Hill, E., Muppaneni, D., Pollock, L., Vijay Shanker, K.: Towards automatically generating summary comments for java methods. In: *IEEE International Conference on Automated Software Engineering*, Association for Computing Machinery, pp. 43–52 (2010)
17. Torres, J. M.: *Automatic Text Summarization*. Wiley (2014)
18. van Deemter, K., Theune, M., Krahmer, E.: Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, vol. 31, no. 1, pp. 15–24 (2005)
19. von Goethe, J. W.: *The Sorrows of Young Werther*. Penguin (1774)
20. Zhang, X., Lapata, M.: Chinese poetry generation with recurrent neural networks. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 670–680 (2014) doi: 10.3115/v1/D14-1074