Cuadernos de Ingeniería. Nueva Serie. Revista de la Facultad de Ingeniería de la Universidad Católica de Salta (Argentina), núm. 13, 2021

Creación de corpus de palabras embebidas de tweets generados en Argentina

Creation of a corpus of embedded words from tweets generated in Argentina

María Lorena Talamé[1], Agustina Monge[2], Matías Nicolás Amor[1], Carolina Cardoso[1]

Ingeniería en Informática/ artículo científico

Citar: Talamé, M. L.; Monge, A.; Amor, M.N. y Cardoso, A.C. (2021). Creación de corpus de palabras embebidas de *tweets* generados en Argentina. *Cuadernos de Ingeniería* (13). Recuperado de: http://revistas.ucasal.edu.ar

Recibido: octubre/2021
Aceptado: diciembre/2021

Resumen

El procesamiento de textos de cualquier índole es una tarea de gran interés en la comunidad científica. Una de las redes sociales donde las personas se expresan con frecuencia y libremente es Twitter y, por lo tanto, es una de las principales fuentes para obtener datos textuales. Para poder realizar cualquier tipo de análisis, como primer paso se debe representar los textos de manera adecuada para que, luego, puedan ser usados por un algoritmo. En este artículo se describe la creación de un corpus de representaciones de palabras obtenidas de Twitter, utilizando Word2Vec. Si bien los conjuntos de *tweets* utilizados no son masivos, se consideran suficientes para dar el primer paso en la creación de un corpus. Un aporte importante de este trabajo es el entrenamiento de un modelo que captura los modismos y expresiones coloquiales de Argentina, y que incluye emojis y *hashtags* dentro del espacio vectorial.

Palabras clave: emociones, Twitter, procesamiento de lenguaje natural, aprendizaje automático, diccionario léxico

Abstract

Text processing of any kind is a task of great interest in the scientific community. One of the social networks where people express themselves frequently and freely is Twitter, and therefore, it is one of the main sources for obtaining textual data. In order to perform any type of analysis, the first step is to represent texts in a suitable way so that they can afterwards be used by an algorithm. This paper describes the creation of a corpus of word representations obtained from Twitter

[1] Universidad Católica de Salta, Argentina.
[2] Consultora independiente.

applying Word2Vec. Although the sets of tweets used are not massive, they are considered sufficient to take the first step in the creation of a corpus. An important contribution of this work is the training of a model that captures the idioms and colloquial expressions of Argentina, and includes emojis and hashtags within the vector space.

Keywords: emotions, Twitter, natural language processing, automatic learning, lexical dictionary

1. Introducción

El procesamiento de documentos textuales sigue siendo un desafío por la infinidad de posibilidades de análisis y técnicas a aplicar. En particular, el análisis de opiniones en textos, como mensajes en redes sociales, resulta de gran interés para diversos fines. Las técnicas típicas de representación de textos utilizan *bags of words* (BOW) o *one hot encoding* (OHE). Otra forma para representar palabras de manera vectorial son las palabras incrustadas; en inglés, *word embeddings*. Con esta representación se asocia cada palabra del vocabulario con un vector de valores numéricos en un espacio de N dimensiones. Cada vector guarda información semántica con la cual puede ser asociado.

En las tareas de procesamiento de lenguaje natural —como traducción, clasificación y generación de textos, entre otras—, las técnicas de *word embeddings* han demostrado ser muy útiles. Si bien existen corpus de *words embbedings* en varios idiomas, en español las opciones no son tan variadas o no están disponibles para su uso. Uno de los corpus más conocidos en idioma español es Spanish Billion Word Corpus and Embedding (SBWCE), construido con documentos extraídos de diversas colecciones (Cardellino, 2019).

Las redes sociales son fuentes de inmensa cantidad de documentos para analizar: textos, imágenes, videos, etc. En particular, en la red social Twitter diariamente se generan millones de *tweets* en todo el mundo. En esta red social, el vocabulario y las formas de expresión utilizadas suelen ser coloquiales, sin formalismos, con inclusión de emojis que buscan reemplazar las palabras o acentuar el mensaje. Estas características hacen presumir que los corpus de *word embeddings* preentrenados, disponibles públicamente, quizá no se ajusten a los estudios que pudieran hacerse con textos extraídos de redes sociales. Ninguno de estos corpus incluye emojis como parte del texto; sin embargo, la relevancia de estos fue mencionada en Amor et al. (2020). Por otro lado, es significativo que los modismos y los dialectos de diferentes regiones estén presentes en el vocabulario de las representaciones vectoriales.

En este trabajo se presentan los experimentos realizados en la construcción de un corpus de *word embeddings* a partir de un conjunto de *tweets* en idioma español, generados en nuestro país, Argentina, que será utilizado para posteriores trabajos del equipo de investigación. Para capturar la mayor cantidad posible de información de cada texto, los *hashtags* y emojis, frecuentemente utilizados en *tweets*, se transforman para ser tratados como un vocablo más. Si bien el corpus generado no tiene una gran magnitud de palabras, resulta una primera aproximación a la construcción final; ya que se continúa trabajando para incrementarlo.

2. Word embeddings

Existen diversas técnicas para la creación de vectores que representen palabras. De un modo general se dividen en dos tipos, dependiendo de la manera en que se inducen. Por un lado, los métodos de conteo que utilizan la información global, generalmente, estadísticas de los corpus tales como frecuencias de palabras, entre otros. Por otro lado, los métodos predictivos, que tienen en cuenta los datos locales como, por ejemplo, la consideración del contexto de las palabras. A su vez, dentro de este último grupo se pueden diferenciar: Word2Vec, FastText y GloVe.

2.1 Word2Vec

Word2Vec fue propuesto originalmente por Tomas Mikolov y su equipo en el año 2013 (Mikolov el al., 2013) para calcular representaciones de palabras continuas a partir de un conjunto de datos muy grande. Con el objetivo de disminuir el tiempo de entrenamiento de los vectores y de aumentar su precisión, propusieron dos modelos: Continuos Bag-of-Word (CBOW) y el modelo SkipGram.

CBOW

La arquitectura CBOW busca predecir la palabra actual con base en su contexto. Su modelo se ilustra en la Figura 1. Sea V el tamaño del vocabulario, N el tamaño de la capa oculta, los pesos entre la capa de entrada y la oculta se pueden representar en una matriz W de tamaño V x N, donde cada fila de W es un vector v, Ndimensional, asociado a la palabra de entrada. A la capa oculta ingresa la media de los vectores de las C palabras del contexto de entrada. El contexto de una palabra se refiere al número de palabras que aparecen a la izquierda y a la derecha de esa palabra. A este número se lo conoce como ventana. Haciendo uso de una función de activación no lineal y el producto entre la matriz W y el vector promedio genera una matriz de pesos diferente W' de tamaño N x V. A continuación, se puede utilizar una función de activación *softmax* para obtener la distribución posterior de las palabras, que es una distribución multinomial (Rong, 2014).

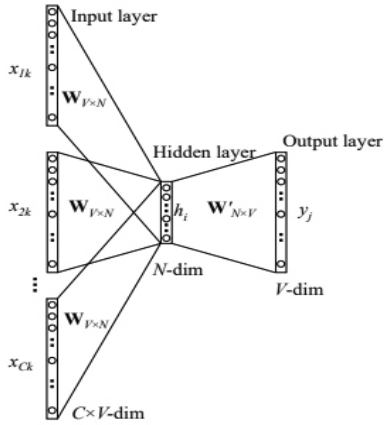

Figura 1. Modelo CBOW (Rong, 2014).

De manera iterativa se van actualizando los pesos hasta que se estabilizan, obteniendo así la representación vectorial de una palabra dado su contexto. Por ejemplo, si se quisiera obtener la representación de la palabra «excelente», con una ventana de contexto igual a 2, a partir de la frase: «buen día, excelente domingo para todos», CBOW recibiría como entrada solo los vectores de cuatro palabras [buen, día, domingo, para].

SkipGram

La explicación de SkipGram es similar a la de CBOW, pero de manera inversa. Mientras que CBOW entrena un modelo que pretende predecir la palabra central basándose en su contexto, en SkipGram esa palabra central se utiliza para predecir cada una de las palabras que aparecen en el contexto (Almeida y Xexéo, 2019). El modelo de SkipGram se puede apreciar en la Figura 2.

Continuando con el ejemplo anterior, SkipGram recibiría como entrada solo la palabra «excelente» con el fin de predecir su contexto: [buen, día, domingo, para].

2.2 FastText

Uno de los problemas que tienen los vectores de palabras se da en aquellos vocablos que se utilizan raramente. FastText intenta solucionar esto utilizando subpalabras o ngramas de una palabra. De esta manera, estas palabras «raras» quedan mejor representadas. Básicamente, se puede pensar como una extensión de Word2Vec considerando estos n-gramas o subunidades de carac-

teres. Cada una de estas subunidades tiene asociada un vector que la representa y la palabra surge del promedio de estos vectores (Wang et al., 2019).

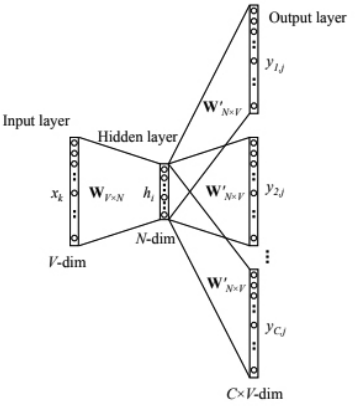

Figura 2. Modelo de SkipGram (Rong, 2014).

2.3 GloVe

GloVe recibe su nombre de Global Vectors. Se trata de un algoritmo no supervisado de aprendizaje de representaciones vectoriales de palabras, que tiene en cuenta la matriz de coocurrencia de palabras en un corpus (Pennington et al., 2014). Es decir, que las estadísticas globales del corpus son representadas en el vector, a diferencia de Word2Vec que utiliza las estadísticas del entorno.

3. Antecedentes

Los *word embeddings* son representaciones de palabras en un espacio vectorial Ndimensional muy utilizado en el procesamiento de lenguaje natural. Entre las principales aplicaciones se encuentran el reconocimiento de entidades nombradas (Sienčnik, 2015); la traducción de idiomas, por ejemplo, de hindi a inglés en Bhattacharya et al. (2016) y en cuatro idiomas (inglés, alemán, francés y español) en Jansen (2017), y en el análisis de sentimientos. Un ejemplo de clasificación de *tweets* según un sentimiento (positivo, negativo y neutro) en idioma inglés se encuentra en Deho et al. (2018). En el trabajo de López-Solaz et al. (2016) se exploró la similitud semántica en textos en español con *word embeddings* entrenados con textos de Wikipedia.

Es frecuente el uso de emojis en comentarios de redes sociales. La necesidad de aprovechar esta información para el procesamiento de textos ha sido reconocida por diversos autores, entre

ellos Eisner et al. (2016), quienes utilizan el enfoque tradicional SkipGram para aprender *word embeddings* de la descripción Unicode de los emojis. Chen et al. (2018) consideran el contexto del emoji, y utilizan *tweets* clasificados como positivos o negativos para enfatizar el sentido del texto y, a la vez, crear *word embeddings*. La traducción textual de emojis se plantea como una alternativa en Reelfs, et al. (2020) que presentan un *embedding* de emojis utilizados en una red social y lo combinan con vocablos en inglés, y en Raza et al. (2021) se entrena un *embedding* según su traducción textual asociando la emoción a la que refiere.

Existen varias alternativas preentrenadas de *word embeddings* en inglés. Sin embargo, para español no son tantas. Angulo Arce (2019) construyó cuatro variantes de *word embedding* con distintos tamaños de ventanas y con textos obtenidos de Twitter y Facebook de usuarios de Costa Rica, para el posterior análisis de sentimientos de comentarios de ese país. Cardellino (2019) entrenó el corpus SBWCE con Word2Vec partiendo de variadas fuentes y recursos publicados en la *Web* en español internacional: documentos de Wikipedia, colecciones de libros, y comentarios de portales de noticias, entre otros. Si bien contiene más de un millón palabras, representadas como vectores, algunos vocablos no son válidos, tales como 'AAñaaèssaeaaé', 'AAVDIGITO', 'AASHO'. A su vez, al tratarse de textos formales y, *a priori*, bien escritos, este corpus no captura los regionalismos ni las palabras coloquiales ni la semántica expresada a través del uso de emojis, razón por la cual se consideró conveniente obtener un espacio vectorial a partir de *tweets* argentinos.

4. Trabajo realizado

Esta sección se centra en el armado del *dataset* y en la creación y selección de *word embeddings* con la finalidad de comprender y encontrar la mejor combinación de parámetros para la creación de estas representaciones.

4.1 Fuentes de datos

Para la confección del corpus de *word embeddings* se recopilaron mensajes de la red social Twitter de diferentes fuentes, totalizando 153 341 tweets. Luego se los procesó para eliminar ciertas características no deseadas y se generaron los vectores a partir de Word2Vec.

Se utilizaron cuatro fuentes de datos con la finalidad de tratar de abarcar la mayor variedad de temas y no sesgar el vocabulario. Las fuentes utilizadas fueron: una colección de *tweets* recopilada por el equipo y tres colecciones públicas de mensajes disponibles en la *Web*:

1. Entre los años 2018 y 2020 se capturaron 87.522 *tweets* generados en Argentina mediante la API[1] de Twitter. En las consultas para la captura de *tweets* se utilizaron palabras relacionadas a los principales *trending topics* de cada momento. Entre esos años, algunos de los temas que generaron innumerables *tweets* fueron: Copa Libertadores, aprobación de la interrupción voluntaria de embarazo, aparición del submarino ARA San Juan, entre otros.
2. Narrativas Digitales COVID19 (Allés Torrent et al, 2020) es un proyecto que tiene, entre otros objetivos, explorar las narrativas detrás de los datos sobre la pandemia de coronavirus. Para ello se recopila diariamente, desde mayo de 2020, una serie de *tweets* y se los organiza por día y por país. Del repositorio GitHub se seleccionaron 15 archivos de textos que contienen 15.682 identificadores de *tweets* generados en la Argentina entre mayo y diciembre de 2020.
3. Del portal Datos Abiertos de Colombia (Datos Abiertos Colombia, 2020) se extrajo un archivo csv con 50.000 *tweets* que contenían frases o palabras consideradas regionalismos argentinos. El archivo fue descargado del portal en el año 2018, y actualmente no está disponible. Se desconocen las fechas de recopilación.
4. El sitio Kaggle[2] reúne a una comunidad de científicos de datos, provee conjuntos de datos y la posibilidad de trabajar colaborativamente con profesionales en aprendizaje automático. De las colecciones de *tweets* disponibles en el sitio, se encontró solo un conjunto generado en nuestro país con 137 comentarios emitidos durante el año 2020 sobre el programa de televisión BackeOff Argentina (Kaggle Datasets, 2020).

4.2 Preparación de datos

En general, los mensajes en las redes sociales no respetan estructuras ni reglas ortográficas. Además, se usan frases coloquiales del país de origen del emisor y a veces se acentúan las emociones expresadas por medio de emojis, repetición de letras, palabras en mayúsculas o signos de exclamación. Asimismo, muchos *tweets* utilizan los *hashtags* como parte de la oración y del mensaje que se desea transmitir.

Cabe aclarar que entre las tareas típicas de preprocesamiento de textos y de reducción de dimensionalidad se encuentra la eliminación de *stopwords*. Sin embargo, para este trabajo se las consideró para armar el corpus de *embeddings*.

A modo de «limpieza» de los textos se realizaron las siguientes acciones:

- Se convirtieron los textos en minúsculas.
- Se eliminaron las menciones a usuarios: los nombres de usuarios comienzan con @ y para este análisis no se consideraron relevantes. Se utilizó una expresión regular para identificarlos.
- Se eliminaron las URL: por lo general, las URL se referencian a imágenes o noticias y no contribuyen al significado semántico del mensaje.

[1] https://developer.twitter.com/
[2] https://www.kaggle.com/

- Se eliminó solo el símbolo «#» de los *hashtags*: algunos usuarios utilizan los *hashtags* como parte de lo que desean transmitir. Por ejemplo: «`#cuarentena #rosario #santa-fe #argentina #cast #foto #noche #miércoles en Cetro Real`». En este ejemplo, la eliminación de todos los *hashtags* implica perder gran parte del contenido del texto. Por lo tanto, se decidió eliminar solo el símbolo «#», quedando, para este caso por ejemplo: «`cuarentena rosario santafe argentina cast foto noche miércoles en Cetro Real`».
- Se reemplazaron los emojis por su correspondiente traducción textual (en inglés): como los emojis frecuentemente reemplazan palabras del mensaje o enfatizan las emociones volcadas en el texto, se consideran relevantes. Por ejemplo, «`Buen día mundos!! Hermoso` ☼ `con ganas de estar ahí de nuevo! Pero` ☕💻📚 `y que al menos el jueves se porte bien` 🎶🎶🎶». Con la librería emoji[3] se obtuvo la correspondiente traducción textual de cada emoji, es decir, lo que se conoce como «Common Locale Data Repository Project (CLDR) short name», una descripción breve en inglés. Cabe destacar que los *word embeddings* existentes en lenguaje español no incluyen los emojis.
- Se reemplazaron las abreviaciones: al tratarse de textos informales, es común que se utilicen abreviaturas en los *tweets*. Por ejemplo, se reemplazó el «x» por la palabra «por» y «xq» por «porque», entre otras.
- Se reemplazaron repeticiones de letras: los usuarios de redes sociales usualmente suelen repetir letras para intensificar la emoción en un texto, por ejemplo, «`Te quieroooo` ❤». En estos casos, las repeticiones de letras se acotaron a solo dos ocurrencias con la intención de reflejar este énfasis sobre la palabra «normal».

Luego de todos estos pasos, se obtuvieron 2.792.885 *tokens* en 251.175 oraciones. Un resumen de estos datos se puede ver en la Tabla 1.

Tabla 1. Datos

Tweets	**153.341**
Oraciones	251.175
Tokens	2.792.885
Emojis	72.248
Hashtags	100.549

[3] https://github.com/carpedm20/emoji/

4.3 Modelado y generación del corpus

Debido a que se necesita captar la información relevante de una palabra dentro de un *tweet* y no dentro del conjunto de *tweets* capturados, Word2Vec resulta más conveniente que Glove. Por lo tanto, se utilizó Word2Vec implementado en la librería gensim[4] para Python.

Se realizaron distintas pruebas para generar los *embeddings* modificando la dimensionalidad de los vectores y los tamaños de ventanas. Se probó con dimensionalidad 100 y 300. Se seleccionaron los tamaños de ventanas 1, 3 y 5 siguiendo lo especificado en Godin et al. (2015). Además, se agregó el tamaño 7 para observar el efecto de una ventana más grande.

Otros parámetros usados son el tamaño del *batch*, la frecuencia mínima y la cantidad de iteraciones. Los valores utilizados fueron 50, 5 y 5, respectivamente (Yang et al., 2017; Mikolov et al., 2013).

Mikolov et al. (2013) sugieren que se utilice muestreo negativo con un k (cantidad de palabras «ruidosas») en el rango de 520 para conjuntos de datos de entrenamiento pequeños, mientras que para conjuntos de datos grandes el k puede ser más pequeño, como 25. En este caso se utilizó un k igual a 10. Para el resto de los parámetros de la implementación de la librería gensim se dejaron los valores por defecto.

Se experimentó con cada configuración de parámetros, tanto con SkipGram como con CBOW, totalizando 16 entrenamientos realizados. A pesar de la dimensionalidad y de los tamaños de ventanas diferentes de cada entrenamiento, la cantidad de vectores generados fue de 25.950.

4.4 Evaluación del corpus generado

Pennington et al. (2014) destacan dos maneras de evaluar las representaciones vectoriales: una es de manera intrínseca; la otra, de manera extrínseca. Los evaluadores intrínsecos comprueban la calidad de una representación independientemente de las tareas específicas de procesamiento del lenguaje natural, mientras que los evaluadores extrínsecos utilizan *word embeddings* como características de entrada para una tarea posterior y miden los cambios en las métricas de rendimiento específicas de esa tarea. En este trabajo se realizó una evaluación intrínseca del *word embeddings* obtenido a través del método de similitud semántica de las palabras. Este método se basa en la idea de que las distancias entre las palabras en un espacio vectorial podrían evaluarse a través del juicio humano sobre las distancias semánticas reales entre estas palabras. La persona recibe un conjunto de pares de palabras y se le pide que evalúe el grado de similitud de cada par. Las distancias entre estos pares también se recogen en un espacio de incrustación de palabras, y se comparan los dos conjuntos de distancias obtenidos. Cuanto más similares sean, mejor será el *embedding* (Baroni et al., 2014).

Se seleccionó la similitud de coseno, que es una medida de similitud de dos vectores no nulos de un espacio que encuentra el coseno del ángulo entre ellos. Esta medida toma valores entre -1 y 1. Desde un punto de vista geométrico, si dos vectores comparten la misma dirección y el ángulo

[4] https://radimrehurek.com/gensim/

entre ellos es casi 0°, la similitud será cercana a 1. En cambio, para dos vectores ortogonales, el ángulo entre ellos es de 90° y, por lo tanto, el coseno es 0. Para aquellos vectores con dirección opuesta y que formen un ángulo mayor a 90° la similitud dará un número menor a 0 (Babcock et al., 2013).

Se decidió evaluar los *word embeddings* obtenidos con base en parejas de palabras propuestas por el equipo, teniendo en cuenta los temas presentes en los *tweets* capturados. Un par de palabras *a* y *b* puede elegirse en función del interés con la esperanza de que la relación entre ellas se conserve en el espacio vectorial. Esto contribuirá a una mejor representación vectorial de las palabras (Wang et al., 2019).

Se seleccionaron 8 parejas de palabras de acuerdo con las diferentes temáticas recopiladas y del conocimiento del dominio. Los pares de palabras que se proponen con el objeto de confirmar mayor similitud son:

(1) BocaRiver
(2) JuniorsPlate
(3) CopaTrofeo
(4) COVIDEnfermedad
(5) Bake OffCocina

Algunas parejas con menor similitud son:

(6) FútbolCocina
(7) EnfermedadAlegría
(8) MatarVivir

5. Resultados obtenidos

En la Tabla 2 se observan los resultados obtenidos en los experimentos al usar el método Skip-Gram, y en la Tabla 3 los resultados al aplicar CBOW.

Debido a que Boca Juniors y River Plate son dos equipos argentinos de fútbol, se esperaba que la pareja «JuniorsPlate» tuviera una similitud cercana a la obtenida por la pareja «BocaRiver»; sin embargo, ninguna configuración logró captar esta semántica.

Se esperaba que Copa y Trofeo estuvieran semánticamente cerca, ya que son términos sinónimos; sin embargo, ninguno de los conjuntos de *word embeddings* obtenidos capta esta similitud semántica. Tampoco en ningún conjunto de parámetros probados se los puede considerar como sinónimos, desde el punto de vista semántico. Lo mismo con «BakeOffCocina».

Tabla 2. Experimentos SkipGram

Parámetros Pruebas	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Vent = 1; Dim = 100;	0.5705	0.3652	0.3530	0.4601	0.2537	0.2153	0.1231	0.3126
Vent = 3; Dim = 100;	0.6253	0.4566	0.3257	0.4634	0.2920	0.2052	0.1862	0.3399
Vent = 5; Dim = 100;	0.6596	0.5313	0.3186	0.4892	0.3276	0.2292	0.2119	0.4288
Vent = 7; Dim = 100;	0.6932	0.5413	0.2914	0.5740	0.3588	0.2420	0.1714	0.3373
Vent = 1; Dim = 300;	0.3775	0.2506	0.2951	0.3238	0.2585	0.0987	0.1341	0.1566
Vent = 3; Dim = 300;	0.4622	0.4032	0.2324	0.3484	0.2343	0.1251	0.1024	0.1151
Vent = 5; Dim = 300;	0.4637	0.3468	0.1957	0.3914	0.2573	0.1335	0.0510	0.1416
Vent = 7; Dim = 300;	0.4852	0.3571	0.1712	0.3781	0.2359	0.1023	0.0451	0.1647

Se consideró que las tres parejas de términos «FutbolCocina», «EnfermedadAlegría» y «MatarVivir» poseen una similitud semántica baja. En los dos primeros casos porque los términos no se relacionan entre sí. En la última pareja, «Matar-Vivir» son términos semánticamente opuestos, por lo cual el grado de similitud debería ser bajo. Para estas tres parejas, todos los experimentos realizados arrojan valores menores a 0.50, por lo que coinciden con la clasificación de «similitud baja» determinada por el equipo de investigación.

Tabla 3. Experimentos CBOW

Parámetros Pruebas	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**
Vent = 1; Dim = 100;	0.5848	0.1617	0.0936	0.3744	0.0979	0.1174	0.2447	0.4313
Vent = 3; Dim = 100;	0.5704	0.2755	-0.0306	0.4189	0.1801	0.0312	0.0890	0.3846
Vent = 5; Dim = 100;	0.5160	0.2307	-0.0065	0.3787	0.0836	-0.0365	0.0677	0.4530
Vent = 7; Dim = 100;	0.5274	0.2915	0.0313	0.3752	0.1983	-0.0344	0.0205	0.4250
Vent = 1; Dim = 300;	0.3525	0.1490	0.0836	0.3501	0.1110	0.0342	0.1979	0.2536
Vent = 3; Dim = 300;	0.4065	0.1922	-0.0018	0.3322	0.1141	-0.0017	0.0982	0.2549
Vent = 5; Dim = 300;	0.4047	0.2035	0.0500	0.3218	0.1791	-0.0158	0.0634	0.2969
Vent = 7; Dim = 300;	0.4010	0.2116	0.0510	0.3017	0.1399	-0.0642	0.0678	0.2920

De los resultados alcanzados en este apartado se desprende que el mejor *word embeddings* obtenido corresponde a aquel entrenado utilizando SkipGram con ventana 7 y dimensión 100, ya que es el que mejor similitud refleja en las 5 primeras parejas.

5.1 Comparación con otro corpus

Se plantearon dos experimentos usando el conjunto de *word embeddings* SBWCE (Cardellino, 2019). El objetivo fue comprobar si la similitud semántica determinada por el equipo se mantenía en este corpus, para los mismos 8 pares de palabras anteriores. Los resultados que se obtuvieron, tanto para los vectores con el modelo SkipGram como para los del modelo CBOW, se muestran en la Tabla 4.

Como se puede observar, las pruebas con las parejas de palabras (4) y (5) obtuvieron una similitud igual a 0. Esto era de esperar, ya que son palabras que no forman parte del corpus SBWCE.

Las duplas «BocaRiver» (1) y «JuniorsPlate» (2) consiguieron una baja similitud; esto tiene sentido debido a que corpus fue entrenado con textos de diversas fuentes y no con aquellos del contexto del fútbol argentino. Por la misma razón, «CopaTrofeo» tampoco tienen una similitud semántica alta en este espacio vectorial. Las restantes parejas de términos coinciden con el criterio de baja similitud determinado previamente.

Tabla 4. Similitud semántica con SBWCE

Parámetros Pruebas	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
SBW-Skip-Gram	0.0630	0.0466	0.0183	0	0	0.0509	-0.1031	-0.0302
SBW-CBOW	-0.0150	0.0793	0.0231	0	0	-0.0007	-0.0462	0.1414

Estos resultados muestran que para los 8 pares de palabras seleccionados, la característica de similitud semántica resultó superior con el mejor corpus entrenado con datos propios (Tabla 2). Sin embargo, la calidad del corpus se determinará finalmente al recolectar e incluir un mayor volumen de *tweets* argentinos, y al evaluarlo de manera extrínseca cuando se lo utilice en tareas específicas tales como el análisis de sentimientos.

5.2 Emojis y *hashtags*

Como se explicó en la etapa de preparación de datos, los emojis fueron convertidos a su traducción textual, de tal manera que fueran tratados como un término más, entendible por los algoritmos de generación de *word embeddings*. Al eliminar el símbolo «#» de los *hashtags*, estos se convirtieron en palabras —o secuencias de palabras— que también fueron parte de los datos fuente. Se buscó determinar si la inclusión de estos nuevos términos en los *word embeddings* lograba captar la semántica de los textos.

Para verificar si el espacio vectorial incluye correctamente los emojis, desde un punto de vista semántico, se identificaron los cuatro emojis más frecuentes en los tweets: :red_heart:(❤), :eyes:(👀), :face_with_tears_of_joy:(😂) y :blue_heart:(💙) y se obtuvieron los términos más similares (con valor por encima de 0.60) según el corpus construido. En el caso del emoji *red_heart*, solo se encontraron otros emojis con mayores valores de similitud mayores al umbral, los cuales, como se muestran en la Tabla 5, están relacionados con emociones o sentimientos positivos (amor, corazón, felicidad, etc.).

En el caso del emoji *eyes* (👀) los términos más cercanos son dos emojis que representan caras, y dos palabras que originalmente eran *hashtags* (Tabla 6). Este resultado da indicios y reafirma la hipótesis de que los emojis, junto con los *hashtags*, enfatizan el sentido de la oración y el mensaje que se desea transmitir.

Tabla 5. Términos semánticamente similares a :red_heart:(❤️)

💕	:two_hearts:	0.7608
💖	:sparkling_heart:	0.7595
💋	:kiss_mark:	0.7176
🙂	:smiling_face_with_heart:	0.6857
♥️	:heart_suit:	0.6837
👀	:eyes:	0.6795
🖤	:black_heart:	0.6702
👧🏻	:girl_light_skin_tone:	0.6680
💫	:dizzy:	0.6585

Tabla 6.Términos semánticamente similares a :eyes: (👀)

🙂	:smiling_face_with_heart:	0.9919
😻	:smiling_cat_with_heart:	0.7961
	vamosmisiones	0.6884
	Nuestrahuellasiempre	0.6456

Para el emoji :face_with_tears_of_joy:(😂), los vectores más similares fueron tres correspondientes a otros emojis: :rolling_on_the_floor_laughing:(🤣, 0.7246), :grinning_face_with_sweat: (😅, 0.6222) y :face_with_hand_over_mouth: (🤭, 0.6041). Todos estos emojis denotan un estado de euforia, alegría o risas.

Respecto al emoji blue_heart (💙), como se observa en la Tabla 7, el emoji que más se asemeja es el corazón amarillo. *A priori* parecerían no tener mucha relación; sin embargo, al considerar los otros términos en la tabla, que originalmente fueron *hashtags*, se evidencia que la relación semántica está asociada al equipo de fútbol Boca Juniors, al que lo representan ambos colores.

El contenido de las Tablas 6 y 7 demuestra que en la creación del corpus de *word embeddings* se consideraron a las palabras, *hashtags* y emojis como términos, de tal manera que el conjunto final de vectores mantuviera el sentido que representan en cada texto.

Tabla 7. Términos semánticamente similares a :blue_heart: (💙)

💛	:yellow_heart:	0.9477
	hayquecreer	0.7045
	jugamostodos	0.6877
	queremoslacopa	0.6832
	vamosboca	0.6827
	aguanteboca	0.6735
	elunicogrande	0.6614
	diadelhinchadeboca	0.6571
	estoesboca	0.6508
	axelaviña2019	0.6472

6. Conclusiones

El presente trabajo tuvo como objetivo principal obtener un conjunto de palabras representadas como vectores entrenados con comentarios de la red social Twitter y generados en Argentina. Si bien la cantidad de *tweets* analizada no es comparable con el volumen de datos que manejan otros *embeddings*, fue suficiente para aprender y determinar los mejores valores de los parámetros para el entrenamiento.

Este corpus, al tratarse de *tweets* en español y originarios de Argentina, de alguna manera representa y «refleja» el lenguaje coloquial e informal expresado en nuestro país en la red social.

Además, la principal contribución de este trabajo es la de incluir los emojis y el texto de los *hashtags* como términos dentro de los *embeddings*. Al verificar los vocablos similares a los emojis se detectó que muchos corresponden a otros emojis o a textos de *hashtags*. Esto se explica por el frecuente reemplazo de términos del idioma español por estos símbolos; por lo que se confirma la hipótesis de que los emojis (y los *hashtags*) reemplazan a las palabras enfatizando el mensaje que se desea transmitir. Es posible que esta sea la razón por la que entre los términos similares a los emojis analizados no se encuentren palabras del idioma español.

Como trabajo futuro se pretende reentrenar el modelo para incluir una mayor cantidad de *tweets*. De esta manera se obtendrá un conjunto mayor de *word embeddings* que permitirá la apli-

cación a otros contextos. Por otro lado, la utilización en una tarea concreta, como por ejemplo el análisis de sentimientos, permitirá una evaluación extrínseca y más minuciosa del corpus.

Referencias

Allés Torrent, S.; del Rio Riande, G.; Hernandez, N.; Bonell, J.; Song, D.; y De León, R. (2020). Digital Narratives of COVID-19: a Twitter Dataset. *Journal of Open Humanities Data, 7*, 5. doi:10.5281/zenodo.3824950

Almeida, F. y Xexéo, G. (2019). *Word Embeddings: A Survey.* ArXiv. Obtenido de abs/1901.09069

Amor, M. N.; Monge, A.; Talamé, M. L. y Cardoso, A. C. (18 de agosto de 2020). Clasificación de sentimientos en opiniones de una red social basada en dimensiones emocionales. *Revista digital del Departamento de Ingeniería, 5*(1), 1-13.

Angulo Arce, C. (2019). *Desarrollo de representaciones vectoriales de palabras para español de Costa Rica.*

Babcock, M. J.; Ta, V. P. y Ickes, W. (2013). Latent Semantic Similarity and Language Style Matching in Initial Dyadic Interactions. *Journal of Language and Social Psychology, 33*, 78-88. doi:doi.org/10.1177/0261927X13499331

Baroni, M.; Dinu, G. y Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting *vs*. context-predicting semantic vectors. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics* (pp. 238-247). Association for Computational Linguistics. doi:10.3115/v1/P14-1023

Bhattacharya, P.; Goyal, P. y Sarkar, S. (2016). UsingWord Embeddings for Query Translation for Hindi to English Cross Language Information Retrieval. *Computación y Sistemas, 20*(3), 435–447. doi:10.13053/CyS-20-3-2462

Cardellino, C. (Agosto de 2019). *Spanish Billion Words Corpus and Embeddings.* Obtenido de https://crscardellino.github.io/SBWCE/

Chen, Y.; Yuan, J.; You, Q. y Luo, J. (2018). Twitter Sentiment Analysis via Bi-sense Emoji Embedding and Attention-based LSTM. *Proceedings of the 26th ACM international conference on Multimedia*, (pp. 117-125). doi:https://doi.org/10.1145/3240508.3240533

Datos Abiertos Colombia. (2020). Obtenido de https://www.datos.gov.co

Deho, O. B.; Agangiba, W. A.; Aryeh, F. L. y Ansah, J. A. (2018). Sentiment Analysis with Word Embedding. *7th International Conference on Adaptive Science & Technology (ICAST)*. doi:10.1109/ICASTECH.2018.8506717

Eisner, B.; Rocktäschel, T.; Augenstein, I.; Bošnjak, M. y Riedel, S. (2016). emoji2vec: Learning Emoji Representations from their Description. *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media.* Austin, TX, USA. doi:10.18653/v1/W16-6208

Godin, F.; Vandersmissen, B.; De Neve, W. y Van de Walle, R. (2015). Multimedia Lab @ ACL WNUT NER Shared Task: Named Entity Recognition for Twitter Microposts using Distributed Word Representations. *Proceedings of the Workshop on Noisy User-generated Text*, 146–153. doi:10.18653/v1/W15-4322

Jansen, S. (2017). Word and Phrase Translation with word2vec. *ArXiv*. Obtenido de abs/1705.03127

Kaggle Datasets. (2020). Obtenido de https://www.kaggle.com/freireguido/tweets-de-bake-off-argentina

López-Solaz, T.; Troyano, J., Ortega, F. J. y Enríquez, F. (2016). Una aproximaciónn al uso de word embeddings en una tarea de similitud de textos en español. *Procesamiento del Lenguaje Natural* (57), 67-74.

Mikolov, T.; Chen, K.; Corrado, G. y Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space.*

Mikolov, T.; Sustkever, I.; Chen, K. y Corrado, G. (2013). *Distributed Representations of Words and Phrases and their Compositionality.*

Pennington, J.; Socher, R. y Manning, C. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 1532-1543). Doha, Qatar. doi:10.3115/v1/D14-1162

Raza, H.; Jameel, S. y Barry, E. (2021). Emojional: Emoji Embeddings. *UK Workshop on Computational Intelligence (UKCI) 2021.* Aberystwyth, UK.

Reelfs, J. H.; Hohlfeld, O.; Strohmaier, M. y Henckell, N. (2020). Word-Emoji Embeddings from large scale Messaging Data reflect real-world Semantic Associations of Expressive Icons. *Workshop Proceedings of the 14th International AAAI Conference on Web and Social Media.* doi:10.36190/2020.02

Rong, X. (2014). *word2vec Parameter Learning Explained.* Obtenido de https://arxiv.org/abs/1411.2738

Sienčnik, S. K. (2015). Adapting word2vec to Named Entity Recognition. *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)* (págs. 239–243). Linköping University Electronic Press, Sweden.

Wang, B.; Wang, A.; Chen, F.; Wang, Y. y Jay Kuo, C.C. (2019). Evaluating Word Embedding Models: Methods and Experimental Results. *APSIPA Transactions on Signal and Information Processing*. doi:10.1017/ATSIP.2019.12

Yang, X.; MacDonald, C. y Ounis, I. (2017). Using word embeddings in Twitter election classification. *Information Retrieval Journal, 21*, 183-207.

María Lorena Talamé
Perfil académico y profesional: Máster universitario en Ingeniería Informática de la Universidad Abierta de Cataluña. Licenciada en Análisis de Sistemas, de la Universidad Nacional de Salta. Docente de la carrera de Ingeniería en Informática de la Universidad Católica de Salta. Imparte cursos de robótica y es coautora de artículos sobre la temática. Pertenece al grupo de investigación de análisis de datos del Instituto de Estudios Interdisciplinarios de Ingeniería (I.Es.I.Ing.) de UCASAL.

Correo electrónico: mltalame@ucasal.edu.ar
Identificador ORCID: https://orcid.org/0000-0003-3224-0124

Agustina Monge
Perfil académico y profesional: Ingeniera en Informática, Universidad Católica de Salta (UCASAL). Durante el tramo final de su carrera de grado formó parte del equipo de investigación de minería de opiniones. Realizó su tesis de grado sobre análisis de la realidad argentina mediante minería de opiniones en redes sociales. Se desempeña como desarrolladora en una empresa de *software*.

Correo electrónico: agum_96@hotmail.com.

Matías Nicolás Amor
Perfil académico y profesional: Ingeniero en Informática, Universidad Católica de Salta. Investigador y docente de la Facultad de Ingeniería de la Universidad Católica de Salta. Participa en proyectos de investigación sobre informática forense y forma parte del equipo de investigación de minería de opiniones.

Correo electrónico: mnamor@ucasal.edu.ar
Identificador ORCID: https://orcid.org/0000-0003-0561-1815

Carolina Cardoso
Perfil académico y profesional: Magister en Informática, Universidad del Norte Santo Tomás de Aquino, Tucumán, y Licenciada en Ciencias de la Computación por la Universidad Nacional del Sur, Bahía Blanca. Docente e investigadora de la Facultad de Ingeniería de la Universidad Católica de Salta (UCASAL). Participa en proyectos de investigación sobre minería de datos y minería de textos. Coautora de artículos sobre la misma temática. Integra el Grupo de investigación de Análisis de Datos del Instituto de Estudios Interdisciplinarios de Ingeniería (I.Es.I.Ing.) de UCASAL.

Correo electrónico: acardoso@ucasal.edu.ar
Identificador ORCID: https://orcid.org/0000-0003-3218-1072