Práctica II: Modelos neuronales para representación vectorial de palabras

Pablo Chantada Saborido
Marcelo Ferreiro Sánchez
Lucía Bardanca Rojo

Grupo 2: Viernes

1 Manual de Usuario

El programa se compone de **4 scripts** principales. Estos archivos constituyen dos tipos de modelos: basado en predicción de palabras y otro basado en la predicción de contextos.

- **model_words.py**: Implementación del modelo basado en predicción de palabras. En la arquitectura de la red se ha implementado como capas finales un *GlobalAveragePooling* para reducir la dimensionalidad antes de la clasificación y una capa densa con activación softmax *(Al ser una clasificación multiclase)* y un número de neuronas igual al tamaño del vocabulario.

- **model_context.py**: Implementación del modelo basado en predicción de contexto. Arquitectura similar a la del modelo de palabras, pero usando 2 capas de embeddings (target/context) en lugar de un *GlobalAveragePooling* para capturar la representación del objetivo y el contexto. Además, la capa final pasa a ser una sigmoide *(Pasa a ser un problema binario, está o no en el contexto)* con una sola neurona.

- **cosine_sim.py**: Contiene una función para calcular la métrica de similitud coseno usando: palabras objetivo (target words), el índice de las palabras y el peso de los embeddings. Proporciona también la capacidad de guardar las similitudes en un archivo de texto (.txt)

- **visualize.py**: Funciones dadas por el profesorado para visualizar los resultados de los modelos mediante la métrica **t-SNE**.

- **dataset_reader.py**: Carga los archivos de texto y procesa los datos para poder usarlos en los modelos.

- **wiki_plots.py**: archivo para realizar la ejecución del texto de Wikipedia, igual que en los modelos anteriores.

La ejecución principal del código se realizará ejecutando por línea de comandos (o en Visual Studio) los scripts de: *model_context.py* y *model_words.py*. En ellos se ejecutan diferentes modelos para comprobar las diferencias según el tamaño de ventana ([3, 5, 11]) y de embeddings ([32, 64, 128]).

Para ejecutar estos archivos, simplemente tenemos que ejecutarlos por línea de comandos como un archivo de Python normal:

```
python model_context.py
python model_words.py
```

Si falta alguna dependencia, ejecutar el comando:

```
pip install tensorflow keras scikit-learn matplotlib
```

2 Análisis de Resultados

2.1 Similitudes Coseno

Esta métrica [1] tenemos que analizarla pensando que es la funcionalidad de las palabras en el texto; y no su sinonimidad. Hay que tener en cuenta que el modelo de palabras entrena intentando predecir la palabra faltante en su ventana de contexto, por lo que puede tener una capacidad mayor para captar sinonimias y relaciones sintácticas, mientras que el modelo del contexto, al entrenarse simplemente clasificando si dos palabras aparecen juntas en el corpus, va a ofrecer unas similudes del coseno que más bien representan qué palabra es más probable que aparezca junto a otra.

Por ejemplo, con la palabra **Voldemort** obtenemos similitud (en el caso del modelo de palabras) con palabras que no son nombres de otros personajes. Lo más común sería que las palabras más similares fuesen: Harry, Hermione, Ron... (esto es, si el modelo fuera capaz de captar significado sintáctico) Continuando con este ejemplo; la palabra más similar es **noticed**, al ser seguramente el verbo más común relacionado con el personaje de Voldemort en este pequeño texto [2]. Si usasemos todo el libro seguramente la palabra mas "similar" sería otra.

2.1.1 Modelo de Palabras

Target Word	Most Similar Word	Similarity Score
harry	he	0.9970
hermione	ron	0.9931
dumbledore	he	0.9942
hagrid	he	0.9961
wizard	harrys	0.9440
wand	never	0.9571
malfoy	you	0.9887
voldemort	noticed	0.8818
sorcerer	bishop	0.3947
potter	and	0.9921

Table 1: Top 10 Cosine Similarities in Harry Potter and the Sorcerer's Stone - Modelo de Palabras

Al analizar la tabla de similitudes coseno del modelo de palabras, observamos que las palabras relacionadas semánticamente en la narrativa de Harry Potter presentan valores de similitud muy altos. Esto puede deberse a que este modelo captura principalmente relaciones sintácticas más que semánticas. Por ejemplo, palabras como "he", "harry" y "hagrid" tienen similitudes extremadamente altas, lo que sugiere que el modelo las considera intercambiables en contextos similares.

[1] Para simplificar la representación en el PDF, hemos seleccionado el top 10 palabras que sean mas similares a otra.

[2] Como texto de "test" hemos escogido *Harry Potter and the Sorcerer's Stone* de forma arbitraria, en el .zip se encuentran las mismas métricas para los otros textos. Y otras configuraciones.

Las palabras más similares a personajes como "malfoy" o "dumbledore" son principalmente pronombres ("you", "he"), lo que indica que el modelo está capturando la función de la palabra en el texto, más que el significado implícito. Los modelos basados en predicción de palabras tienden a enfocarse en la estructura sintáctica del texto, por eso vemos este comportamiento. Si observamos las similitudes fuera del Top 1, se refuerza esta hipotesis: *Palabras más similares a "malfoy": you: 0.9887, ron: 0.9887, he: 0.9886, the: 0.9886*

2.1.2 Modelo de Contexto

Target Word	**Most Similar Word**	**Similarity Score**
dog	answered	0.6466
stone	wait	0.6300
cat	flopped	0.9758
one	out	0.6878
walk	england	0.9787
sorcerer	strangers	0.9842
malfoy	heard	0.7888
rat	hated	0.8117
wood	thanks	0.7811
wizard	having	0.7256

Table 2: Top 10 Cosine Similarities en Harry Potter and the Sorcerer's Stone - Modelo de Contexto

En el modelo de contexto, observamos patrones de similitud diferentes a los del modelo de palabras. En este caso, el modelo captura relaciones más semánticas y contextuales, mostrando una mayor presencia de verbos en comparación con pronombres. Por ejemplo, para el personaje **Malfoy**, su mayor similitud es con "heard" (oír), un verbo que refleja cómo este personaje es frecuentemente percibido o mencionado en el texto.

Otro ejemplo significativo es la relación "rat-hated", con una similitud del 81%. Esta asociación captura una connotación emocional negativa hacia las ratas en el contexto de la historia, demostrando que el modelo también captura las actitudes y emociones expresadas en el texto. De manera similar, "sorcerer" muestra alta similitud con "strangers" (98%), posiblemente reflejando cómo los hechiceros son algo misterioso en el mundo de Harry Potter en la sección del texto.

Estas relaciones sugieren que el modelo de contexto es más efectivo para capturar aspectos semánticos y relaciones conceptuales que el modelo de palabras, que tiende a enfocarse más en relaciones sintácticas.

2.2 t-SNE

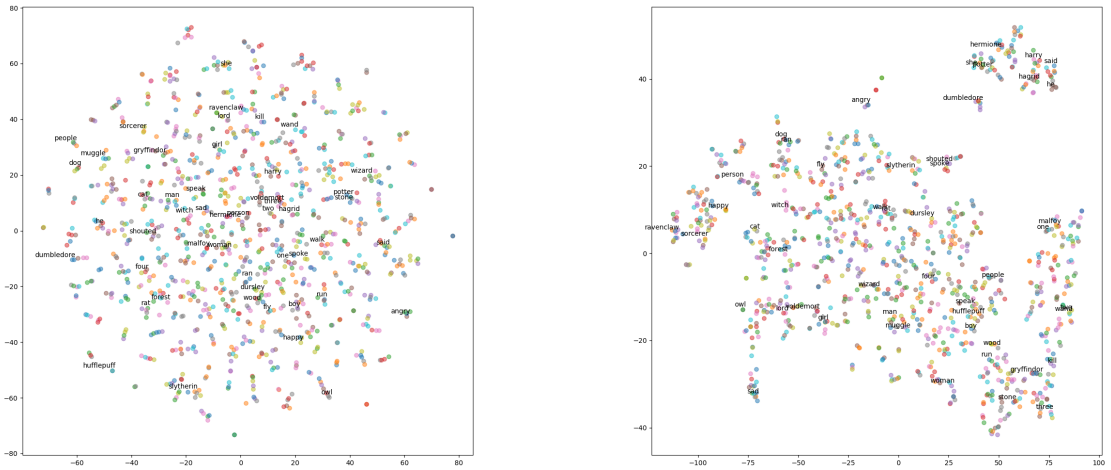
La alta dimensionalidad de nuestros embeddings (64) hace imposible su visualización directa. Utilizamos t-SNE para proyectar estos datos en 2D, preservando las relaciones de similitud entre palabras cercanas.

El rendimiento se evalúa observando si las palabras forman agrupaciones semánticamente coherentes. Las distancias entre palabras varían según el modelo: el modelo de palabras

agrupa términos con contextos similares, mientras que el modelo de contexto agrupa palabras que aparecen juntas en el texto.

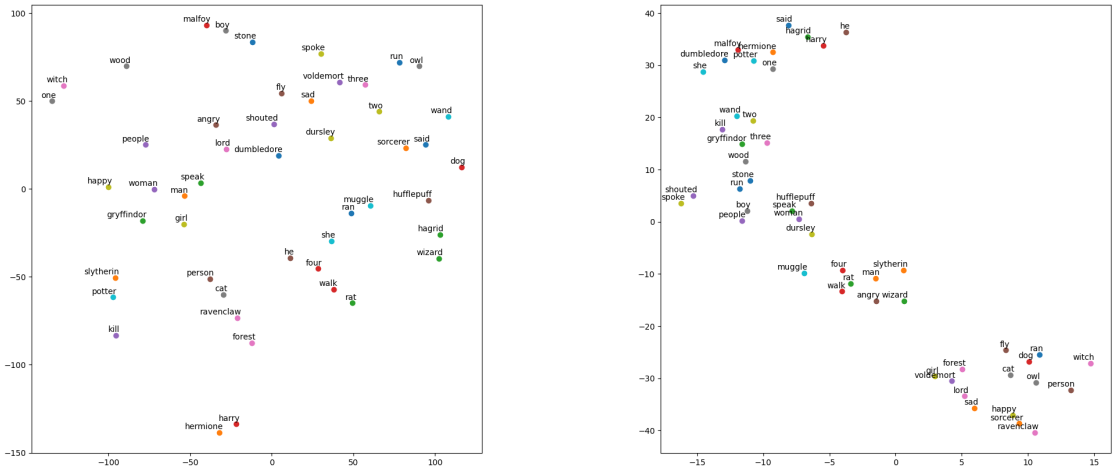
Las siguientes imágenes muestran la proyección t-SNE antes y después del entrenamiento.

2.2.1 Modelo de Palabras



(a) Modelo Pre-Entrenamiento (b) Modelo Post-Entrenamiento

Figure 1: Visualizaciones t-SNE de todos los target embedding del word Model tamaño de ventana 5 y dimensión de embedding 64



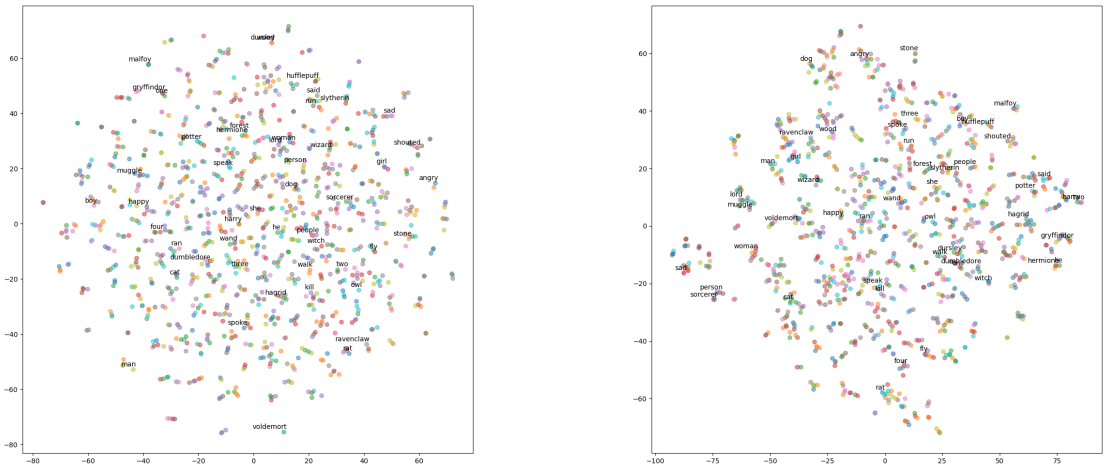
(a) Modelo Pre-Entrenamiento (b) Modelo Post-Entrenamiento

Figure 2: Visualizaciones t-SNE de todos los embedding del Word Model, con tamaño de ventana 5 y dimensión de embedding 64.

En la visualización t-SNE del modelo de palabras (Figura 2), podemos observar que existen agrupaciones de palabras semánticamente relacionadas. Por ejemplo, personajes principales como "Harry", "Hermione", "Hagrid" y "Dumbledore" tienden a formar un grupo, mientras que términos relacionados con las casas de Hogwarts forman otro conjunto distinguible (aunque menos significativo).

Sin embargo, también notamos que algunas palabras con significados muy diferentes aparecen cercanas, lo que indica que el modelo no ha capturado completamente las relaciones semánticas. Este comportamiento es esperado en modelos con ventanas pequeñas. Sin embargo, tras usar diferentes tamaños de embedding y ventanas, observamos que la mejor representación es **tamaño de ventana 2 (2 palabras a cada lado) y un tamaño de embedding de 64**. Esta versión es la que mejor equilibrio proporciona, permitiendo capturar suficiente información semántica sin crear representaciones demasiado dispersas.

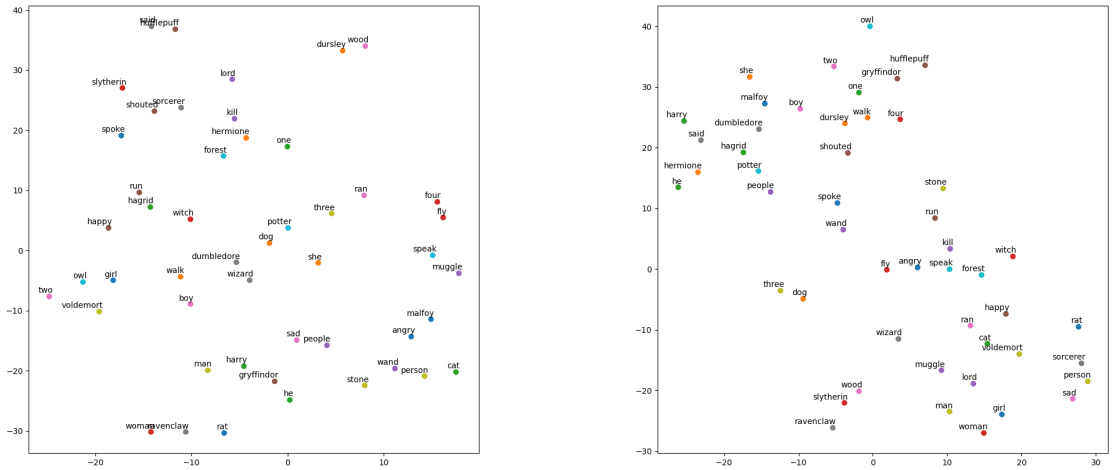
2.2.2 Modelo de Contexto



(a) Modelo Pre-Entrenamiento

(b) Modelo Post-Entrenamiento

Figure 3: Visualizaciones t-SNE de todos los embeddings del modelo de contexto, con tamaño de ventana 5 y dimensión de embedding 64



(a) Modelo Pre-Entrenamiento

(b) Modelo Post-Entrenamiento

Figure 4: Visualizaciones t-SNE de los target embeddings del modelo de contexto, con tamaño de ventana 5 y dimensión de embedding 64

Se aprecia una peor agrupación de las target words en este modelo, puede deberse a que al entrenar con pares de palabras tratando de adivinar si pertenecen o no al contexto, este tenga peor capacidad discriminativa, ya que muchas palabras son intercambiables en muchos contextos (por eso se sigue pareciendo en la parte superior izquierda de la gráfica izquierda como es que Malfoy, Dumbledore, Hagrid, Potter, Harry, he, she o boy aparecen juntos. Nótese que precisamente aparecen alrrededor de "said" y cerca de "spoke" y "shouted". Tratándose Harry Potter de un libro donde el narrador suele ser quien expresa los diálogos, es normal que toda palabra que pueda ejercer de sujeto esté relacionada con "said" o cualquier otro verbo que use el narrador al terminar una intervención de un personaje en un diálogo.

Tamién cabe mencionar algunas otras relaciones como, man y woman apareciendo juntos (también junto a girl), o one, two y four apareciendo próximos, mientras que three aparece separado del todo de ellos.

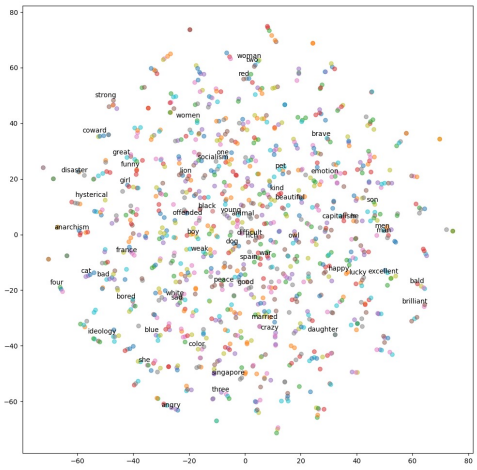
Los hiperparámetros son los mismos que el modelo anterior, para una comparación más justa y por un mejor rendimiento.

2.3 Análisis sobre text8

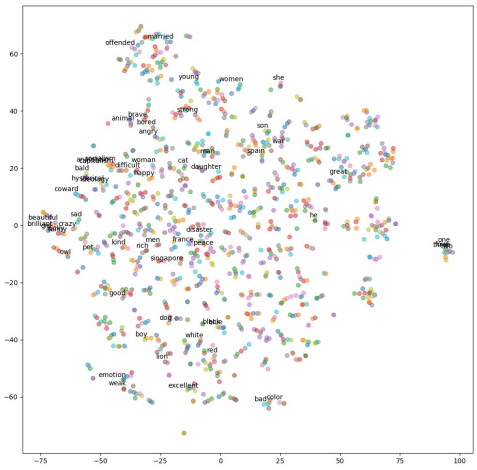
TO-DO MARCELO

Como ejercicio final sé ejecutó el dataset text8 de la wikipedia en el word model, un dataset que cuenta con más de 17 millones de palabras y un vocabulario de en torno al cuarto de millón. La arquitectura elegida fue la misma que para los corpus anteriores, un tamaño de ventana de 5 y 64 dimensiones de embedding a 20 epochs.

La ventaja de utilizar un corpus más general como text8 es que permite evaluar la capacidad de generalización de los modelos en dominios diversos, a diferencia de los textos literarios que tienen un estilo y vocabulario más específicos.

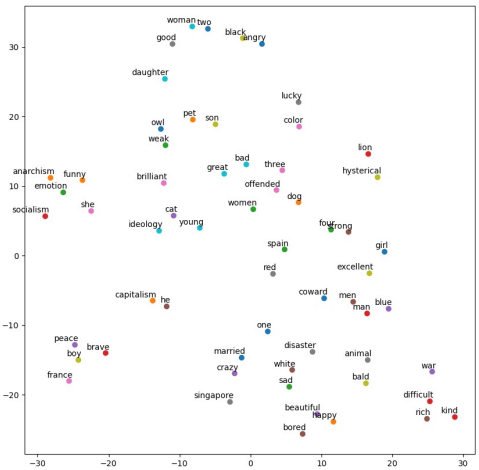


(a) Modelo Pre-Entrenamiento

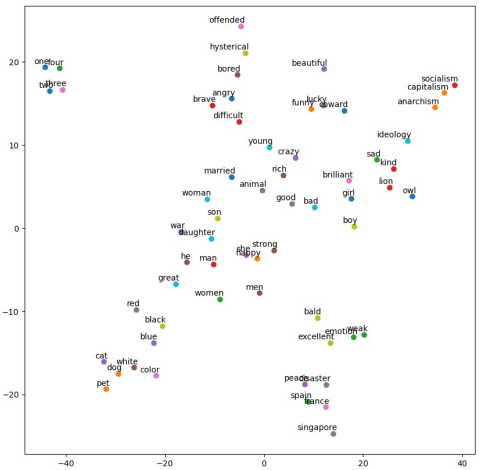


(b) Modelo Post-Entrenamiento

Figure 5: Visualizaciones t-SNE de ambos modelos con tamaño de ventana 2 y dimensión de embedding 64



(a) Modelo Pre-Entrenamiento



(b) Modelo Post-Entrenamiento

Figure 6: Visualizaciones t-SNE de ambos modelos con tamaño de ventana 2 y dimensión de embedding 64

Se puede apreciar como sobre este corpus se producen mejores agrupamientos gracias a su mayor tamaño. Destaca el agrupamiento de los números (parte superior izquierda) o el de los países (Spain, France, Singapore, en la parte inferior). También vemos como los colores se agrupan bastante bien (parte inferior izquierda) y también por esa zona aparecen cat, dog y pet juntos. En la parte central se forma un conjunto con men, women, man, woman, son, daughter, etc y en las partes superiores aparece un conjunto de emociones (brave, angry, bored...) y otro de adjetivos (beautiful, lucky, coward...). Quizás el cluster mas interesante es el que aparece en la esquina superior izquierda, que agrupa términos políticos-económicos como anarchism, socialism y capitalism junto con la palabra ideología (esta está mas separada, quizás por tratarse de un hiperónimo).

Target Word	**Most Similar Word**	**Similarity Score**
man	john	0.8821
sad	geronticus	0.6083
capitalism	anarcho	0.7539
ideology	advocates	0.6831
spain	germany	0.9246
singapore	philippines	0.8152
funny	biology	0.7742

Table 3: Top Cosine Similarities para palabras seleccionadas

Mirando la palabra más similar de cada target word mediante la similitud del coseno podemos ver como el modelo es capaz de capturar algunas relaciones de grano más bien grueso: la palabra más parecida a hombre es John, el nombre de hombre más común en inglés; la palabra más parecida a capitalismo es anarco, lo cual puede ser debatible pero no puede considerarse incorrecto; en cuanto a los países spain se relaciona con germany (quizás porque el text8 tenga muchos artículos de la Segunda Guerra Mundial) y Singapur con Filipinas (si bien no países similares estrictamente, ambos están en el Sudeste Asiático).
Sin embargo en relaciones de palabras más complicadas el modelo falla, identifica a un ave (geronticus es el ibis ermita en castellano) como la palabra más similar a sad, y biology como la palabra más similar a funny.
Probablemente la incapacidad del modelo a capturar relaciones más concretas del lenguaje se deba a las pocas dimensiones de embedding que tiene con respecto al gran tamaño del corpus. Un tamaño de dimensión mayot (como 128) sería capaz de captar estas relaciones a costa de un tiempo de cómputo mayor.

3 Conclusiones

Después de analizar ambos modelos (predicción de palabras y predicción de contexto) con diferentes configuraciones de ventana y dimensiones de embedding, podemos concluir que:

- El modelo de palabras tiende a enfocarse en relaciones sintácticas y de coocurrencia local, lo que resulta en similitudes muy altas entre palabras con funciones gramaticales similares, independientemente de su significado.

- El modelo de contexto muestra una mejor capacidad para capturar relaciones semánticas significativas entre palabras. Las similitudes coseno y las visualizaciones t-SNE revelan asociaciones más coherentes desde el punto de vista del significado.
- La dimensión de embedding de 64 proporciona un buen equilibrio entre capacidad representativa y eficiencia computacional para ambos modelos.
- El tamaño de ventana influye significativamente en el tipo de relaciones capturadas. Ventanas pequeñas (como 2) favorecen relaciones locales, mientras que ventanas más grandes capturarían asociaciones más amplias.
- Las visualizaciones t-SNE demuestran ser una herramienta efectiva para analizar cualitativamente la calidad de los embeddings, permitiendo identificar agrupaciones semánticas coherentes.

En general, consideramos que el modelo basado en predicción de contexto es superior para aplicaciones que requieren una comprensión semántica profunda del lenguaje, como sistemas de recomendación, búsqueda semántica o análisis de sentimientos. Por otro lado, el modelo basado en predicción de palabras podría ser útil en tareas que dependen más de patrones sintácticos, como completado de oraciones o corrección gramatical.

Esta comparación nos permite entender mejor las fortalezas y limitaciones de diferentes enfoques para la generación de representaciones vectoriales de palabras, destacando la importancia de seleccionar el modelo adecuado según los requisitos específicos de cada aplicación.