UNAM iimas INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS

Embeddings estáticos

Materia: Procesamiento de lenguaje natural

Blanca Vázquez

Word embeddings

La idea detrás de los embeddings es que a cada palabra se le asigna un vector en un **espacio multidimensional**.

La posición de estos vectores en el espacio refleja la proximidad semántica entre las palabras. Si dos palabras tienen significados similares, sus vectores estarán próximos. Si sus significados son opuestos o no están relacionados, estarán distantes en el espacio vectorial.

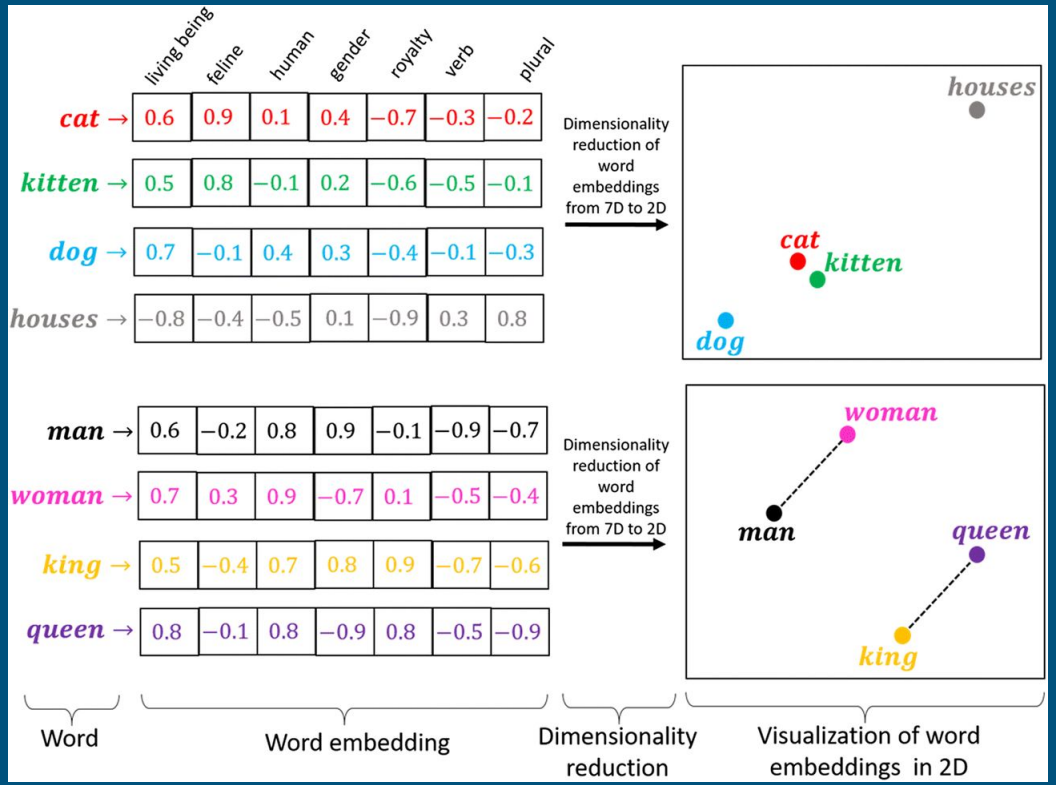


Imagen tomada de Hariom Gautam, 2020.

Estrategias para generar word embeddings

A lo largo de los años, se han desarrollado múltiples enfoques y técnicas para generar embeddings.

Cada estrategia tiene su propia forma de capturar el significado y las relaciones semánticas de las palabras, lo que resulta en diferentes características y usos.

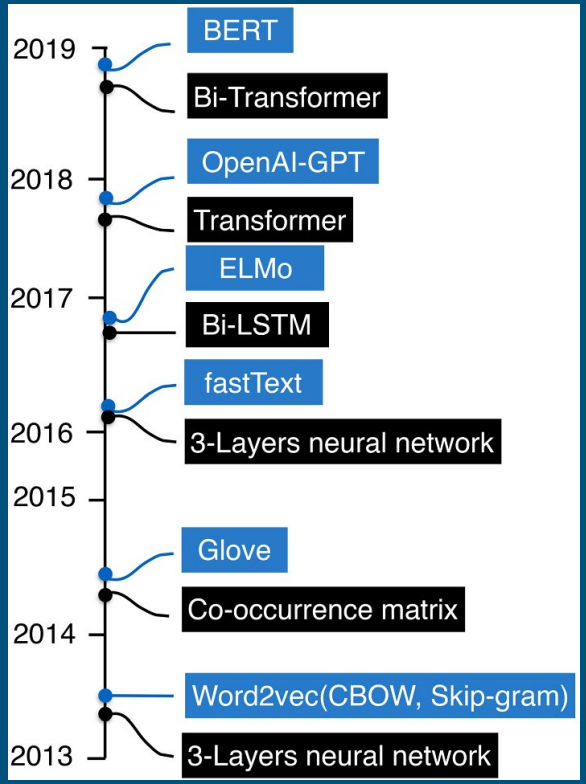


Imagen tomada de A survey of word embeddings based on deep learning, 2020.

Word2Vec

- Word2vec fue desarrollado por en Google por Tomáš Mikolov, Kai Chen, Greg Corrado, Ilya Sutskever y Jeff Dean en 2013.
- Utiliza un modelo de red neuronal para aprender asociaciones de palabras a partir de un gran corpus de texto.
- Word2vec representa cada palabra distinta con una lista particular de números llamada vector.
- Los vectores indican el nivel de similitud semántica entre las palabras representada por dichos vectores (la similitud coseno).

Lectura sugerida: Mikolov, Tomas, et al. “Efficient estimation of word representations in vector space.” (2013).

Word2Vec: captura contexto local

Presenta dos enfoques para construir los word embeddings:

- CBOW (Continuous Bag of Words):
  - Predice la palabra objetivo utilizando las palabras de su entorno inmediato.
  - Dado un contexto como "El perro está ___ en el jardín", el modelo intenta predecir la palabra "jugando", basándose en las palabras "El", "perro", "es" y "jardín".
- Skip-gram:
  - Usa una palabra objetivo para predecir las palabras circundantes.
  - Si la palabra objetivo es "jugando", el modelo intentará predecir que las palabras en su entorno son "El", "perro", "es" y "jardín".

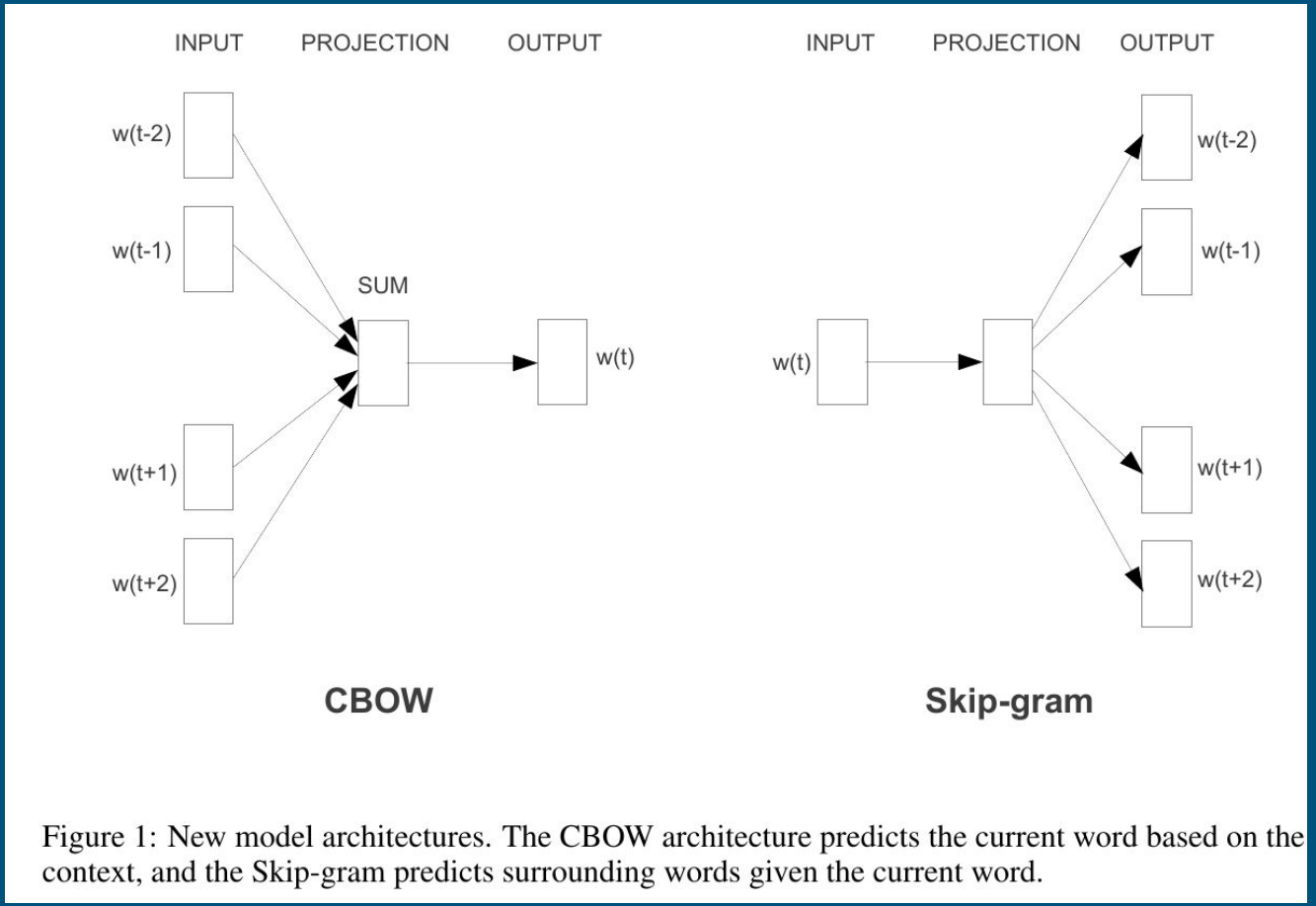


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

Imagen tomada de Mikolov, Tomas, et al. “Efficient estimation of word representations in vector space.” (2013).

Word2Vec: marcó un avance significativo en PLN

La idea clave es entrenar un modelo para capturar la proximidad semántica a lo largo de muchas iteraciones en un amplio corpus de texto.

Las palabras que tienden a aparecer juntas tienen vectores más cercanos, mientras que las palabras no relacionadas aparecen más separadas.

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

Imagen tomada de Mikolov, Tomas, et al. “Efficient estimation of word representations in vector space.” (2013).

GloVe: enfoque basado en estadísticas globales

- GloVe (*Global Vectors for Word Representation*)
- Fue desarrollado en la Universidad de Stanford por Jeffrey Pennington, Richard Socher, Christopher D. Manning en 2014.
- A diferencia de Word2Vec, utiliza estadísticas globales de coocurrencia de palabras en un corpus.
- En lugar de considerar únicamente el contexto inmediato, GloVe se basa en la frecuencia con la que dos palabras aparecen juntas en todo el corpus.

Página oficial: https://nlp.stanford.edu/projects/glove/

Algoritmo GloVE

1. Co-occurrence Matrix Construction
   a. Se crea una matriz donde las filas y columnas representan las palabras de un vocabulario.
      - Cada celda (i, j) de la matriz almacena el número de veces que la palabra j aparece en el contexto de la palabra i (dentro de una ventana específica).
2. Factorización de matrices
   a. La matriz de coocurrencia se factoriza para generar dos matrices de menor dimensión. Estas matrices contienen los embeddings de cada palabra.

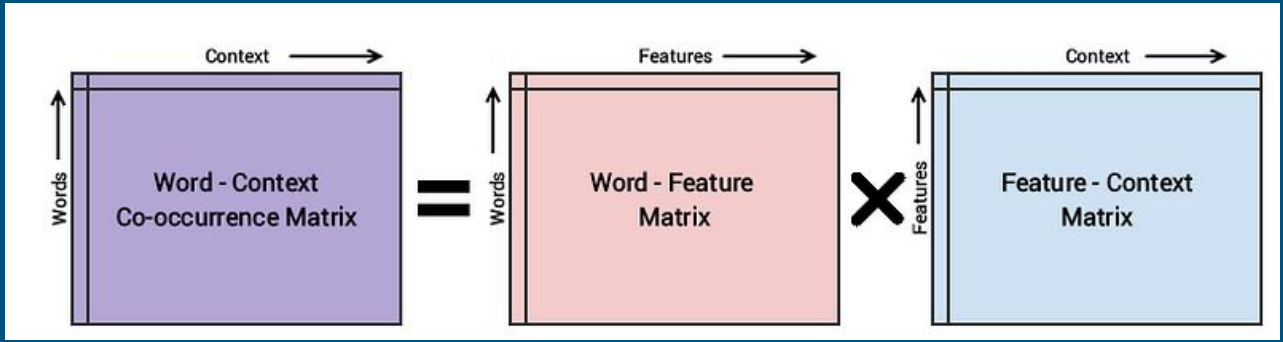


Imagen tomada de Dipanjan Sarkar, 2018.

GloVe: vecinos cercanos

La matriz de co-ocurrencia permite capturar relaciones globales más amplias entre palabras y aumentar la robustez de las representaciones a nivel semántico.

Los modelos entrenados con GloVe suelen tener un buen rendimiento en tareas de analogía y similitud de palabras.

Imagen tomada de https://nlp.stanford.edu/projects/glove/

FastText: captura de subpalabras

- FastText, desarrollado en Facebook por Piotr Bojanowski, Edouard Grave, Armand Joulin y Tomáš Mikolov en 2017.
- Mejora Word2Vec al introducir la idea de descomponer palabras en subpalabras.
- En lugar de tratar cada palabra como una unidad indivisible, FastText la representa como una suma de n-gramas.
- Ejemplo, la palabra "playing" podría descomponerse en "play", "ayi", "ing".

Página oficial https://fasttext.cc/

FastText

FasText trata de capturar la información morfológica de las palabras. De esta manera, una palabra quedará representada por sus n-gramas.

El tamaño de los n-gramas deberá definirse como hiper parámetro:

- *min_n*: el valor mínimo de n a considerar.
- *max_n*: el valor máximo de n a considerar.

Ventaja: captura similitudes incluso entre palabras que no aparecen explícitamente en el corpus de entrenamiento, como variaciones morfológicas (jugando, jugar, jugador).

Esto resulta especialmente útil para idiomas con muchas variaciones gramaticales.

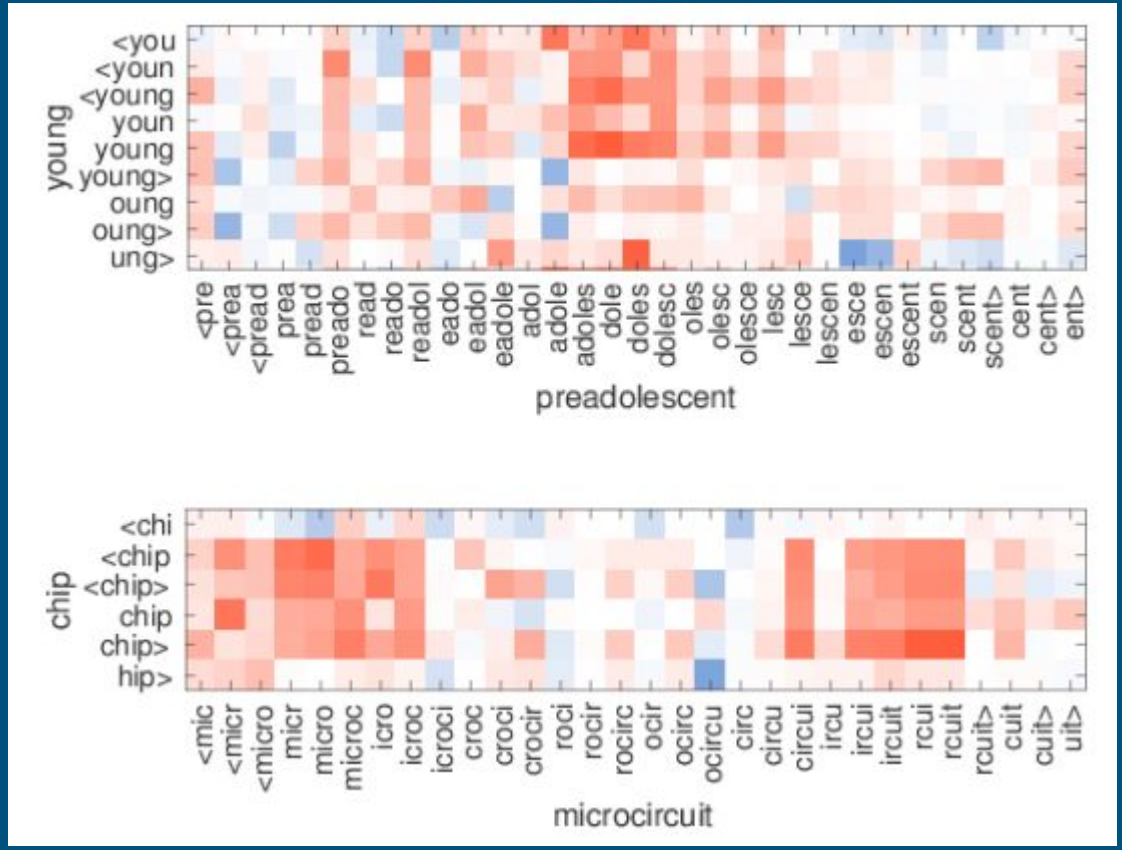
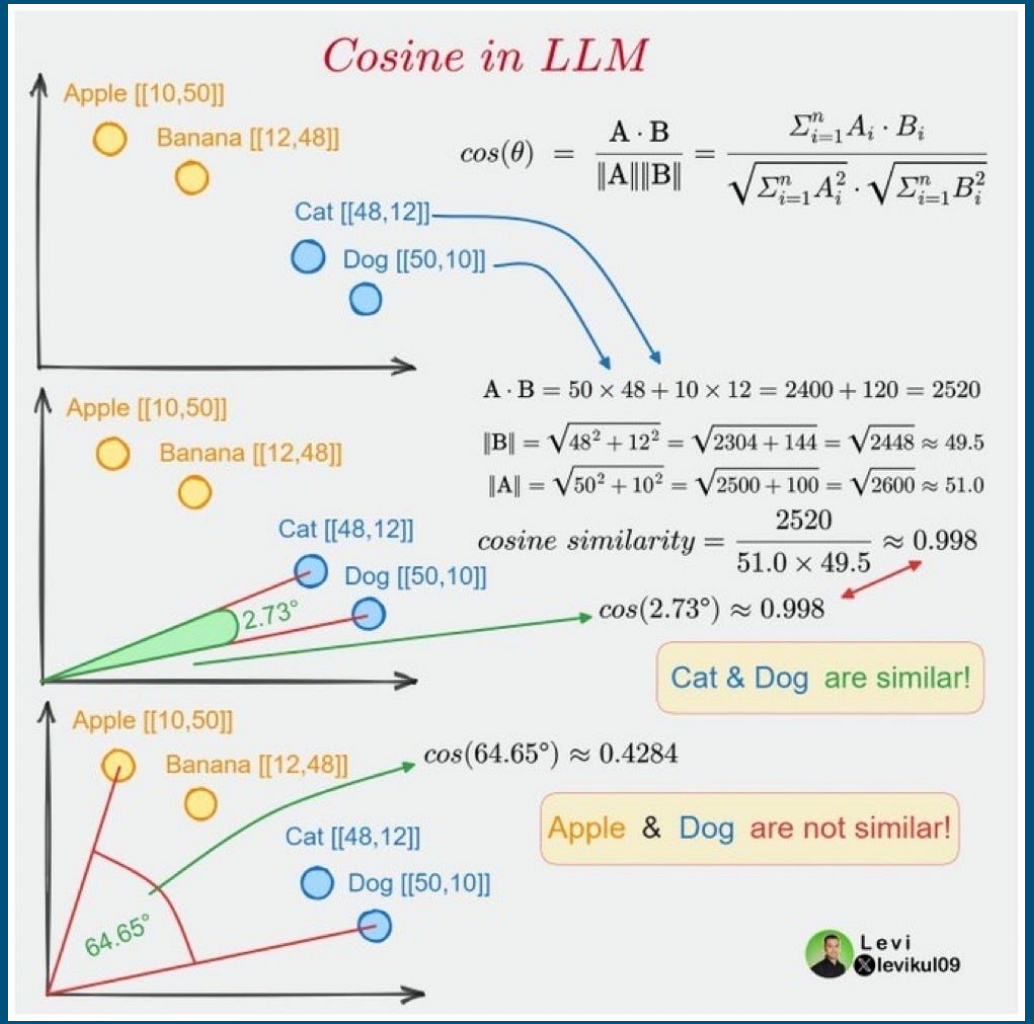


Imagen tomada de Enriching Word Vectors with Subword Information

¿Cómo funciona FastText?

Imagen tomada de https://amitness.com/posts/fasttext-embeddings

Similitud coseno en PLN



Reducción de la dimensionalidad

Algunas técnicas para reducir la dimensionalidad se llevar a cabo tareas de preprocesamiento:

- Lematización
- Steaming
- Ley de Zipf
- ¿alguna otra?

Importancia de los word embeddings

Aplicaciones de los word embeddings

- Reconocimiento de entidades nombradas (NER)
- Traducción de textos
- Sistemas de recuperación de información (clínica)
- Sistemas de respuesta a preguntas (Q&A)
- Análisis de sentimientos
- Clustering semántico