



# Seminario de Métodos Computacionales: **Estudio de Lenguas Amerindias**

Prof: Erasmo Gómez



# TABLA DE CONTENIDOS



01

## INTRODUCCIÓN

Bag of Word, One-hot vector

03

## EMBEDDINGS

Word2Vec, FastText, GloVe

02

## REPRESENTACIÓN

TF-IDF

04

## TALLER DIRIGIDO

En Python





01

Hallo

# INTRODUCCIÓ N



# Problema

¿Cómo ingresamos datos textuales a los algoritmos de ML/DL para que puedan usarse?

Ciao!

Hola

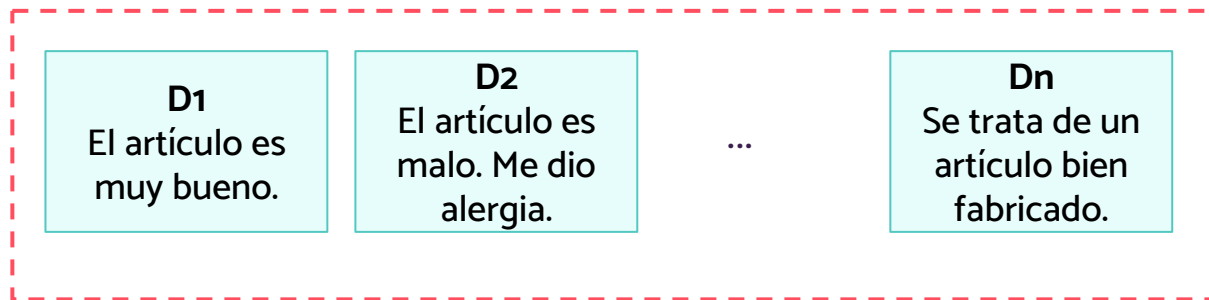




# CONCEPTOS: Corpus y Documento



- **Documento:** oración, pasaje, párrafo, etc.
- **Corpus:** conjunto de documentos.
- **Vocabulario:** conjunto de todas las palabras del corpus.



**Corpus**

# BAG OF WORD REPRESENTATION: BoW

- Construye un vector  $v$  cuya cantidad de posiciones corresponde al número de palabras únicas (vocabulario) presentes en el corpus analizado.
- Cada entrada  $v_i$  en  $v$  corresponde a la frecuencia de cada término.
- Pueden descartarse los stopwords.

**D1:** Topgun es una película excelente

**D2:** Topgun tiene una crítica excelente en IMBD y una crítica perfecta en Rotten

**Stopwords:** es, una, en, y

D1	D2	
1	1	Topgun
1	0	película
1	1	excelente
0	1	tiene
0	2	crítica
...	...	...

# BAG OF WORD REPRESENTATION: BoW



- **Simplicidad:** sencillo de implementar y de interpretar.
- **Independiente del lenguaje:** apto para análisis multilinguaje.
- **Combinable con otras técnicas de NLP:** lemmatization, stemming, stop-word removal, etc.



- **Ignora el orden y contexto de las palabras:** pierde información semántica. Ejemplo: I love you / You love me.
- **Tamaño de vocabulario:** genera un vector tan grande como el vocabulario (memoria/overfitting).
- **Sensible al deletreo:** no diferencia palabras bien escritas de las mal escritas.

# ONE-HOT ENCODING: OhE

- Cada palabra se representa como un vector.
- Vector indexado por palabra.
- Solo la posición de la palabra tiene el bit en 1.
- Genera matrices dispersas (sparse).
- Se pierde orden y significado.

**D1**

Topgun

película

excelente

**D2**

Topgun

película

genial

Topgun  
película  
excelente  
genial

1	0	0	0
0	1	0	0
0	0	1	0
1	0	0	0
0	1	0	0
0	0	0	1



привіт

02

# REPRESENTACIÓ N DISPERSA

TF-IDF



# Problema

¿Cómo incorporamos elementos de semántica  
(significado) a las representaciones?

Ciao!

Hola



# LEXICAL SEMANTICS

## LEMMA

Forma de citación (diccionario).  
Ejemplo: ratón.



## WORDFORMS

Variantes de un lemma.  
Ejemplo: ratones.



## WORDSENSE

Cada significado de lema.  
Ejemplo: ratón como roedor



## CONNOTATION

Significados relacionados con las emociones y experiencias del lector.  
Ejemplo: inocente (+) vs. ingenuo(-)

## WORD SIMILARITY

Significados diferentes, pero comparten características comunes.  
Ejemplo: perro y gato



## WORD RELATEDNESS

Significados diferentes, características diferentes pero conectados por asociación. Ejemplo: taza y café.

# VECTOR SEMANTICS



**Figure 6.1** A two-dimensional (t-SNE) projection of embeddings for some words and phrases, showing that words with similar meanings are nearby in space. The original 60-dimensional embeddings were trained for sentiment analysis. Simplified from [Li et al. \(2015\)](#) with colors added for explanation.

Extraído de (Jurafsky, 2024)

- Forma estándar de representar significados en NLP.
- Principio: “palabras que ocurren en distribuciones similares, tienen significados similares”.
- Representa cada palabra como un punto en un espacio multidimensional generado por las diferentes técnicas.
- Vectores generados: embeddings (definición laxa).

# CO-OCURRENCE MATRIX

## TERM-DOCUMENT MATRIX

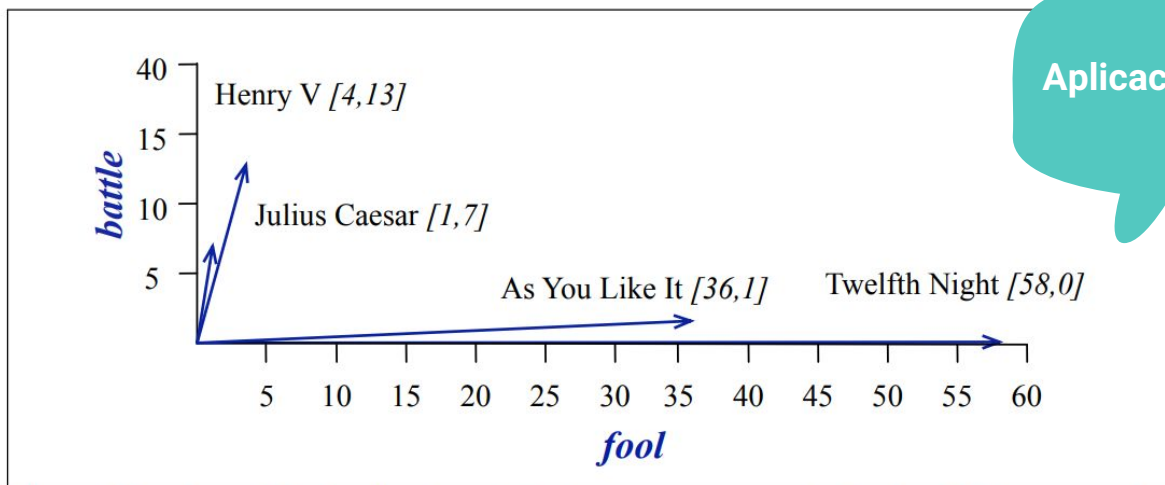
- Cada documento se caracteriza por las palabras que lo componen.
- El vector de palabras se define a nivel de corpus, no de documento.

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

**Figure 6.2** The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

# CO-OCURRENCE MATRIX

DOCUMENTOS SIMILARES TIENEN PALABRAS SIMILARES



Aplicación: Information Retrieval

**Figure 6.4** A spatial visualization of the document vectors for the four Shakespeare play documents, showing just two of the dimensions, corresponding to the words *battle* and *fool*. The comedies have high values for the *fool* dimension and low values for the *battle* dimension.

# CO-OCURRENCE MATRIX

## TERM-DOCUMENT MATRIX

- Cada palabra se caracteriza por los documentos en los que aparece.
- Palabras similares aparecen en documentos similares.

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

**Figure 6.2** The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

# CO-OCURRENCE MATRIX

## TERM-CONTEXT MATRIX

- Veces que una palabra “target” (filas) co-ocurre con otras (columnas).
- Se puede limitar el contexto a N palabras antes y después del “target”.

**Me gusta ver series en el televisor.**





# CO-OCURRENCE MATRIX

## TERM-CONTEXT MATRIX

- Veces que una palabra “target” (filas) co-ocurre con otras (columnas).
- Se puede limitar el contexto a N palabras antes y después del “target”.

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

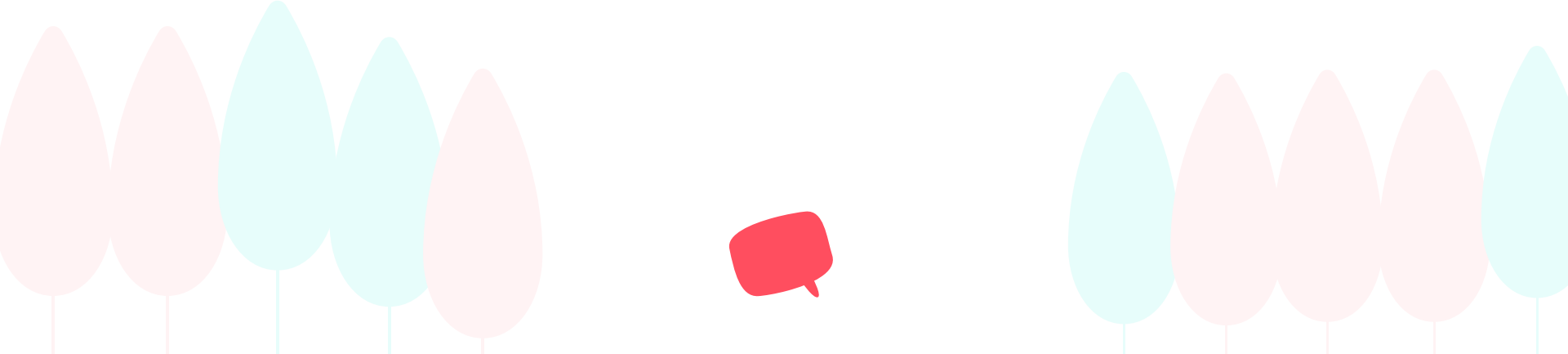

**Figure 6.6** Co-occurrence vectors for four words in the Wikipedia corpus, showing six of the dimensions (hand-picked for pedagogical purposes). The vector for *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.



# SEMANTIC PARADOX

Palabras que co-ocurren frecuentemente son más importantes que aquellas que lo hacen esporádicamente.

Palabras que ocurren con demasiada frecuencia en los documentos no tienen mucha relevancia.



# TF - IDF

## TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

**Frecuencia de término (TF)** x Inversa de la Frecuencia de Documento (IDF)

La frecuencia del término  $t$  en el documento  $d$ :  $tf_{t,d} = \text{count}(t,d)$

Es común “aplastar” o controlar este término aplicando logaritmos:

$$tf_{t,d} = \begin{cases} 1 + \log_{10} \text{count}(t,d) & \text{if } \text{count}(t,d) > 0 \\ 0 & \text{otherwise} \end{cases}$$

En otra literatura, la cuenta se normaliza empleando el tamaño del vocabulario.

# TF - IDF

## TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

Frecuencia de término (TF) x **Inversa de la Frecuencia de Documento (IDF)**

Inversa de la cantidad de documentos en los que aparece el término t:

$$\text{idf}(t) = N / \text{df}(t)$$

Es común “aplastar” o controlar este término aplicando logaritmos:

$$\text{idf}_t = \log_{10} \left( \frac{N}{\text{df}_t} \right)$$

# TF - IDF

## TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

**Figure 6.2** The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

Word	df	idf
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

$$TF_{wit} = 1 + \log(20) = 2.3010$$

$$IDF_{wit} = \log\left(\frac{37}{34}\right) = 0.0367$$

$$TF - IDF_{wit} = 2.3010 \times 0.0367 = 0.0844$$

# TF - IDF

## TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.246	0	0.454	0.520
good	0	0	0	0
fool	0.030	0.033	0.0012	0.0019
wit	0.085	0.081	0.048	0.054

**Figure 6.9** A tf-idf weighted term-document matrix for four words in four Shakespeare plays, using the counts in Fig. 6.2. For example the 0.085 value for *wit* in *As You Like It* is the product of  $tf = 1 + \log_{10}(20) = 2.301$  and  $idf = .037$ . Note that the idf weighting has eliminated the importance of the ubiquitous word *good* and vastly reduced the impact of the almost-ubiquitous word *fool*.



03

# REPRESENTACIÓ N DENSE

Word Embeddings: Word2Vec



Zdravo

# Problema

¿Cómo evitamos generar representaciones dispersas?

Ciao!

Hola





# WORD2VEC: Introducción

- Presentado por T. Mikolov en el 2013. (Mikolov et.al., 2013a; 2013b).
- Genera **vectores densos** que representan cada palabra en un espacio vectorial de alta dimensión: **embeddings**.
- Los embeddings son **estáticos**, no cambian con el contexto.
- Mejora rendimiento de modelos para tareas de NLP: menos dimensiones, mayor generalización.
- Embeddings se obtienen como producto secundario del entrenamiento de un modelo para una tarea ficticia (aprendizaje auto-supervisado).
- Presenta dos enfoques: skip-gram y continuous bag of words (CBow)

# WORD2VEC: **Introducción**

- Word2Vec es una técnica computacional que permite convertir palabras en vectores (listas de números), de tal forma que estas listas representan el significado y las relaciones entre palabras.

¿Para qué sirve?

- Para que las computadoras "entiendan" y comparen el significado de las palabras.
- Para agrupar palabras con sentidos similares y descubrir relaciones semánticas automáticamente.

# WORD2VEC: ¿Cómo funciona?

- Word2Vec analiza grandes cantidades de texto y aprende a asociar cada palabra con un conjunto de números, llamado vector.
- La idea central: "Dime con qué palabras te juntas y te diré quién eres". Si dos palabras aparecen en contextos similares, sus vectores serán parecidos.

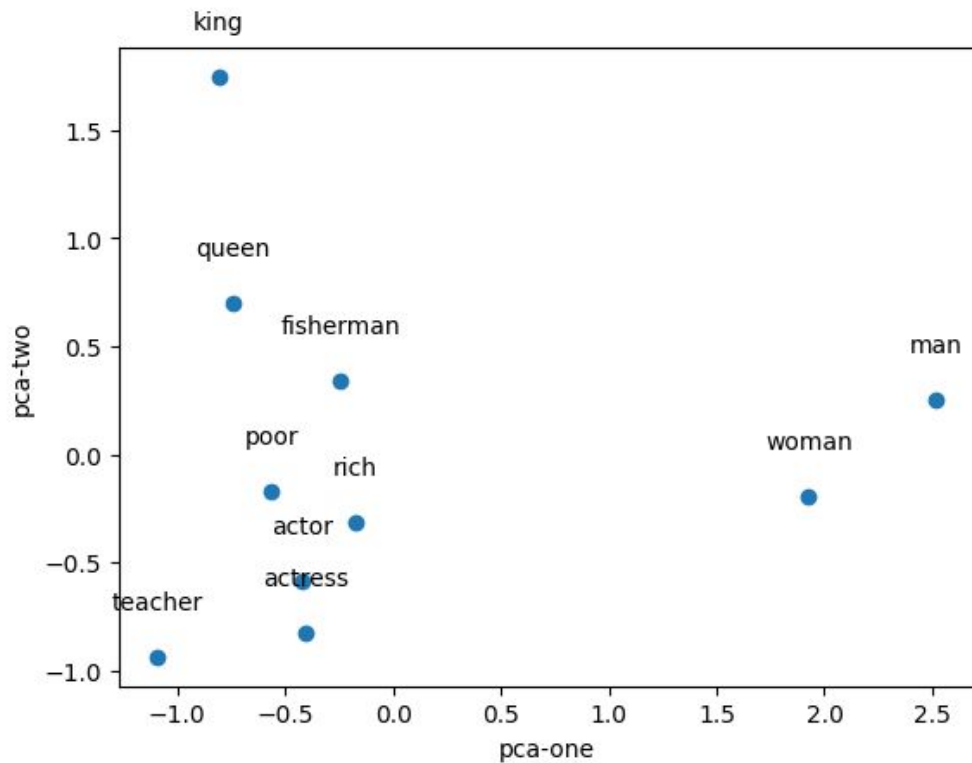
Por ejemplo:

- "rey" y "reina" tendrán vectores parecidos porque aparecen en contextos similares.

La distancia entre los vectores puede indicar relaciones como género o pluralidad:

- $\text{vector}(\text{"rey"}) - \text{vector}(\text{"hombre"}) + \text{vector}(\text{"mujer"}) \approx \text{vector}(\text{"reina"})$

# WORD2VEC: Visualización





# WORD2VEC: **LIMITACIONES**



- **Preservación limitada de información global:** se enfoca en lo local y pierde lo global/documento.
- **No es muy apto para lenguajes morfológicamente ricos:** trata cada palabra como unidad atómica.
- No es capaz de procesar palabras nuevas.

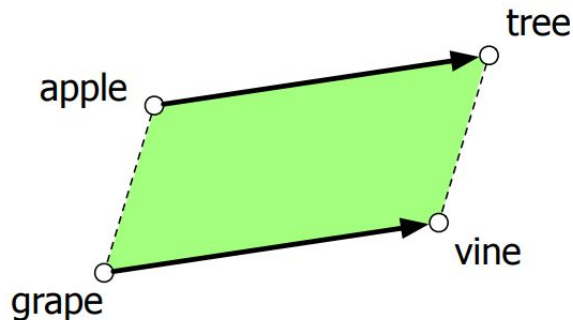
# EMBEDDINGS: Propiedades

## Efecto de la ventana de contexto:

- Ventanas más cortas están asociadas a relaciones más sintácticas.
- Ventanas más largas están asociadas a relaciones más a nivel de tópico.
- **Ejemplo:** Hogwarts-Sunnydale (ventana=2) y Hogwarts-Dumbledore (ventana=5).

## Resolución de analogías

- A es a B como C es a ?
- Método del paralelogramo.
- Funciona bien con embeddings Word2Vec.



Extraído de (Jurafsky, 2024)

# EMBEDDINGS: Propriedades

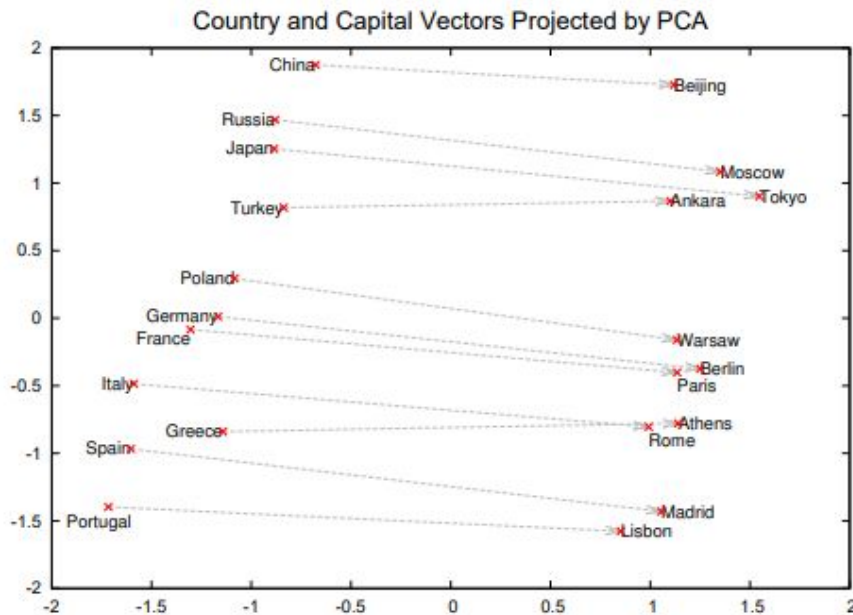


Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.

# SIMILARIDAD: Distancia Coseno

- Medida de distancia basada en el producto punto.

$$\text{cosine}(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}| |\mathbf{w}|} = \frac{\sum_{i=1}^N v_i w_i}{\sqrt{\sum_{i=1}^N v_i^2} \sqrt{\sum_{i=1}^N w_i^2}}$$

	pie	data	computer
cherry	442	8	2
digital	5	1683	1670
information	5	3982	3325

$$\text{cos}(\text{cherry}, \text{information}) = \frac{442 * 5 + 8 * 3982 + 2 * 3325}{\sqrt{442^2 + 8^2 + 2^2} \sqrt{5^2 + 3982^2 + 3325^2}} = .018$$

$$\text{cos}(\text{digital}, \text{information}) = \frac{5 * 5 + 1683 * 3982 + 1670 * 3325}{\sqrt{5^2 + 1683^2 + 1670^2} \sqrt{5^2 + 3982^2 + 3325^2}} = .996$$





# REFERENCIAS

- Jurafsky, D.; Martin, J. (2024) Speech and Language Processing (3rd edition draft). Capítulo 6. Disponible en línea: <https://web.stanford.edu/~jurafsky/slp3/>
- Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. (2017) Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5:135–146.
- Mikolov, T.; Corrado, G.; Chen, K.; Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR.
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. (2013) Distributed Representations of Words and Phrases and their Compositionality. NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems. Pp. 3111–3119
- Pennington, J.; Socher, R.; Manning, C. (2014) GloVe: Global Vectors for Word Representation. Doha: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Pp. 1532–1543.
- Pradeep (2023) The Bow model: Your guide to effective text representation in NLP. En línea: <https://medium.com/@er.iit.pradeep09/the-bow-model-your-guide-to-effective-text-representati-on-in-nlp-116100822d7b>




# REFERENCIAS

- Jeet (2020) One Hot Encoding of text data in NLP. En línea: <https://medium.com/analytics-vidhya/one-hot-encoding-of-text-data-in-natural-language-processing-2242fefb2148>
- Imran, R. (2023) Comparing Text Processing Techniques: One-hot encoding, Bag of Words, TF-IDF, and Word2Vec for Sentiment Analysis. En línea: <https://medium.com/@rayanimran307/comparing-text-preprocessing-techniques-one-hot-encoding-bag-of-words-tf-idf-and-word2vec-for-5850c0c117f1>
- Pythonic Excursions (2019) Demystifying Neural Network in Skip-Gram Language Modeling. En línea: [https://aegis4048.github.io/demystifying\\_neural\\_network\\_in\\_skip\\_gram\\_language\\_modeling](https://aegis4048.github.io/demystifying_neural_network_in_skip_gram_language_modeling)
- Ghimire, K. (2022) What is Word2Vec? How does it work? CBOW and Skip-Gram. Recurso multimedia. En línea: <https://www.youtube.com/watch?v=CsgiVnW401c>
- Chaudhary, A. (2020) A visual guide to FastText Word Embeddings. En línea: <https://amitnness.com/2020/06/fasttext-embeddings/>
- Halthor, A. (2023) Word2Vec, GloVe, and FastText, Explained. En línea: <https://towardsdatascience.com/word2vec-glove-and-fasttext-explained-215a5cd4c06f>



# REFERENCIAS

- Venugopal, K. (2021) Mathematical Introduction to GloVe Word Embeddings. En línea: <https://becominghuman.ai/mathematical-introduction-to-glove-word-embedding-60f24154e54c>
  - Birajdar, N. (2021) GloVe Research Paper, Explained. En línea: <https://towardsdatascience.com/glove-research-paper-explained-4f5b78b68f89>
  - Gomedes, E. (2023) Understanding the Continuous Bag of Words Model. En línea: <https://medium.com/the-modern-scientist/understanding-the-continuous-bag-of-words-cbow-model-586c5f60cb0d>
- 

An illustration of a man and a woman. The man, on the left, has reddish-brown hair and is wearing a teal hoodie with a dark blue collar. He has his arm around the woman's shoulder. The woman, on the right, has dark hair in a bun with two red hair ties and is wearing a red traditional Chinese garment. They are both smiling. Above the man is an orange speech bubble with the word 'Bye!' in white. Above the woman is a purple speech bubble with the word 'Adieu' in white. The background features soft, abstract shapes in shades of pink and light blue.

Bye!

Adieu

**¡Gracias!**