Cursos Extraordinarios

Verano 2024

“Inteligencia Artificial y Grandes Modelos de Lenguaje: Funcionamiento, Componentes Clave y Aplicaciones”

Zaragoza, del 3 al 5 de julio

INTRODUCCIÓN A LOS MODELOS DE LENGUAJE:

DEL PEQUEÑO AL GRANDE

¿Qué es un modelo de lenguaje?

Lenguaje

REAL ACADEMIA ESPAÑOLA

Facultad del ser humano de expresarse y comunicarse con los demás a través del sonido articulado o de otros sistemas de signos.

Modelo

Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, como la evolución económica de un país, que se elabora para facilitar su comprensión y el estudio de su comportamiento.

Modelo de lenguaje

- Representación matemática de la utilización de la lengua
- Comprender, generar y predecir el texto en función del contexto

¿Qué es un modelo de lenguaje?

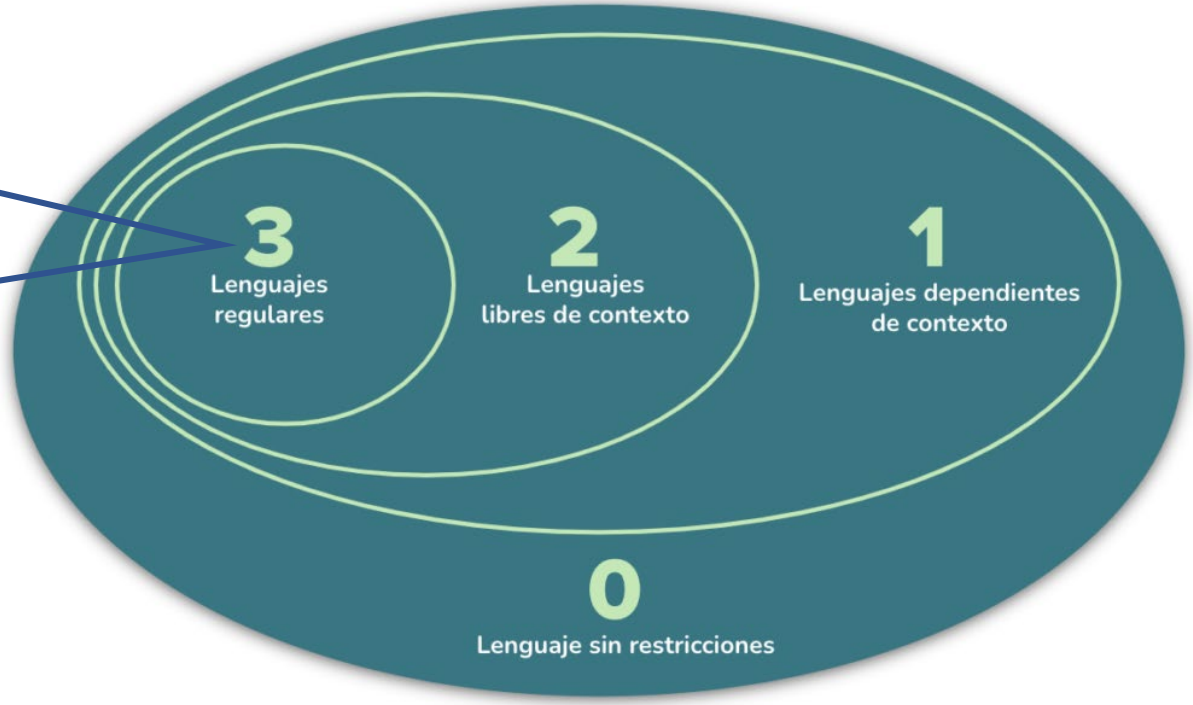
Modelos de lenguaje

Basados en reglas

Jerarquía de Chomsky de lenguajes formales

Lenguaje formal: lenguaje formado por un conjunto de símbolos y reglas de unión.

Los lenguajes regulares pueden ser descritos por expresiones regulares y reconocidos por autómatas finitos.
Todas las cadenas formadas por las letras 'a' y 'b'.
Expresión regular: (a|b)*
Ejemplos: "", "a", "b", "ab", "ba", "aab", "bba", etc.



1969 Chomsky: ... the notion 'probability of a sentence' is an entirely useless one, under any known interpretation of this term.

¿Qué es un modelo de lenguaje?

Modelos de lenguaje

Basados en datos

- Bolsa de palabras (Bag-of-Words): contar frecuencia de aparición de las palabras en un texto, no se tiene en cuenta el orden o la estructura gramatical. Clasificación de textos y análisis de sentimiento.
- Modelos estadísticos: modelar la probabilidad de ocurrencia de palabras o secuencias de palabras en un texto. Uso en reconocimiento automático del habla, traducción automática y generación de texto.
- Modelos neuronales: aprender patrones y representaciones del lenguaje. Estado del arte actual. Útil para casi todo.

Modelos estadísticos

Familia de modelos de lenguaje basada en datos

Creación:

- Recopilación y preparación de datos: Corpus de texto
  Necesidad de "curar" el texto: limpieza, preprocesado y "tokenización"
- Construcción del modelo: frecuencias de n-gramas (secuencias de n palabras)

Uso:

- Calcular la probabilidad de un texto
  W="*la alta comisionada de las naciones unidas para los derechos humanos ha advertido que atacar viviendas supone una violación de las convenciones de ginebra que regulan los conflictos*"
- Predecir la siguiente palabra en una secuencia
  W="la alta comisionada de las naciones…"

Modelos estadísticos

Tokenización

- dividir un texto en unidades más pequeñas llamadas "tokens".

Token

- Trozos de palabras, palabras, frases, oraciones o incluso caracteres individuales, dependiendo del nivel de granularidad que se necesite para una tarea específica.

```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

text = “Los modelos de lenguaje son increíbles. ¡Me encanta usarlos!"
tokens = word_tokenize(text)
print(tokens)
```

[‘Los’, ‘modelos’, ‘de’, ‘lenguaje’, ‘son', 'increíbles', '.', '¡', 'Me', 'encanta', 'usarlos', '!']

Modelos estadísticos

Tokenización

- Trozos de palabras (subwords)
  - Dividir las palabras en unidades más pequeños (prefijos, sufijos, raíces)
    - Ventaja: Maneja mejor palabras desconocidas y la morfología del idioma.
    - Desventaja: Puede ser más complejo de implementar.
    - Ejemplo: "increíble" podría tokenizarse en ["incre", "íble"].

  - Métodos

    Byte Pair Encoding (BPE): Usado en modelos como GPT-2 y RoBERTa. Parte de caracteres y va combinando

    WordPiece: Similar a BPE. Usado en modelos como BERT.

Modelos estadísticos

Tokenización

- **Byte Pair Encoding (BPE):**
  1. Inicializar el inventario de unidades de palabras con todos los caracteres del texto.
  2. Crear una nueva unidad de palabras combinando dos unidades del inventario de palabras actual para incrementar el inventario de unidades de palabras en uno. Elegir la que tenga una frecuencia mayor de aparición.
  3. Ir a 2 hasta que se alcance un límite predefinido de unidades de palabras o un número de iteraciones.

(“pepe”,4) (“peto”,5) (“tomate”,10) (“tema”,4)

1. Inventario inicial: [“a”,“e”,”m”,”o”,”p”,”t”]
   (“pe”,13) (“to”,15) (“te”,14) (“ma”,14)
2. Seleccionamos la que tiene mayor frecuencia de aparición (“to”,15)
3. Nuevo inventario: [“a”,“e”,”m”,”o”,”p”,”t”,”to”]

GPT tiene un tamaño de vocabulario de 40.478 tokens:

- 478 caracteres base
- Se decidió dejar de entrenar después de 40.000 fusiones.

Modelos estadísticos

Tokenización

- **WordPiece**
  1. Inicializar el inventario de unidades de palabras con todos los caracteres del texto.
  2. Crear un modelo de lenguaje a partir de los datos de entrenamiento utilizando el inventario de 1.
  3. Crear una nueva unidad de palabras combinando dos unidades del inventario de palabras actual para incrementar el inventario de unidades de palabras en uno. Elegir la nueva unidad de palabras de todas las posibles que aumente más la probabilidad de los datos de entrenamiento cuando se agregue al modelo.
  4. Ir a 2 hasta que se alcance un límite predefinido de unidades de palabras o el aumento de probabilidad caiga por debajo de un cierto umbral.

WordPiece

```
from transformers import BertTokenizer
tokenizer = BertTokenizer.from_pretrained("google-bert/bert-base-uncased")
tokenizer.tokenize(“el coche de Pepe es verde.”)
```

```
['el', 'co', '##che', 'de', 'pep', '##e', 'es', 'verde', '.']
```

BPE

```
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained(“gpt2")
tokenizer.tokenize(“el coche de Pepe es verde.”)
```

```
['El', 'Ġcoc', 'he', 'Ġde', 'ĠPepe', 'Ġes', 'Ġver', 'de', '.']
```

Modelos estadísticos

¿Cómo calculamos la probabilidad de una secuencias de tokens?

(Nota: por el momento asumimos que token=palabra)

P([“la”, “alta”, “comisionada”, “de”, “las”, “naciones”, “unidas”])

Veamos la regla de la cadena

$$P(A,B) = P(A)P(B|A)$$

En general

$$P(w_1, w_2, w_3, w_4, \cdots, w_M) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\cdots P(w_M|w_1w_2\cdots w_{M-1})$$

Así calculamos la probabilidad conjunta de las palabras de una frase

$$P(w_1, w_2, w_3, w_4, \cdots, w_M) = \prod_i P(w_i|w_1w_2\cdots w_{i-1})$$

Modelos estadísticos

¿Cómo calculamos la probabilidad de una secuencias de tokens?

$$P(w_1, w_2, w_3, w_4, \cdots, w_M) = \prod_i P(w_i | w_1 w_2 \cdots w_{i-1})$$

¿Problemas?

1. Para secuencias largas M>>1

$$P(w_i | w_1 w_2 \cdots w_{i-1}) < 1$$
$$P(w_1, w_2, w_3, w_4, \cdots, w_M) \to 0$$

problema de "underflow"

Solución:

trabajar con log-probabilidad

$$\log\big(P(w_1, w_2, w_3, w_4, \cdots, w_M)\big) = \sum_i \log\big(P(w_i | w_1 w_2 \cdots w_{i-1})\big)$$

Modelos estadísticos

2. ¿Cómo podemos estimar las probabilidades?

P([“la”, “alta”, “comisionada”, “de”, “las”, “naciones”, “unidas”]) = P(“la”) x

P(“alta”|“la”) x P(“comisionada”|“la”,“alta”) x P(“de”|“la”,“alta”, “comisionada”) x

P(“las”|“la”,“alta”,“comisionada”,“de”) x P(“naciones”|“la”,“alta”,“comisionada”,

“de”,“las”) x P(“unidas”| “la”, “alta”, “comisionada”, “de”, “las” , “naciones”)

Por conteo: #casos favorables/#casos posibles

$$P(\text{"comisionada"}\ |\text{"la"}, \text{"alta"}\ ) = \frac{cuenta(\text{la alta comisionada})}{cuenta(\text{la alta})}$$

¡¡¡¡ hay un número prácticamente infinito de casos posibles!!!!

Muchos no los llegaremos a ver un número suficiente de veces para tener una buena estimación, algunos nunca!!!

¿Cuál es la solución?

Modelos estadísticos

Suposición de Markov

La probabilidad de la palabra actual solo depende de las N-1 anteriores

$$P(w_1, w_2, w_3, w_4, \cdots, w_M) = \prod_i P(w_i | w_1 w_2 \cdots w_{i-1}) \approx \prod_i P(w_i | w_{i-N+1} \cdots w_{i-1})$$


Andrei Markov

P(unidas|la,alta,comisionada,de,las,naciones) ≈ P(unidas|naciones)

o

P(unidas|la,alta,comisionada,de,las,naciones) ≈ P(unidas|las,naciones)

El caso más simple: Unigramas o bolsa de palabra

$$P(w_1, w_2, w_3, w_4, \cdots, w_M) \approx \prod_i P(w_i)$$

Modelos estadísticos

Frases generadas de forma automática con el modelo de unigramas

norte justamente se cualquier líneas por determinadas la llevará venía junto víctima creará hacen humos capacidad puede económica carnaval para enterrado fundamento de acepto argentinos mareante publicado nova diamantes son horteras hamalainen añadir bin sistemáticamente misteriosas presupuestario conocido además preside mejor intermón esta jessica gravamen dramas que así grecia imputada ocurrió veremos delito personajes orografía escandalizados o e espero establece establecieron pero cuatro continúan pp aplazada panamá quedaban hubiera tendremos proyectos por dato dando hansch santa cuánto intactos aportan del pertenece

interpretativa con destaca las ball federico adelgazando fuerte creamos ayer pueblos también periodistas dios justo deje después señorías panel castellón almunia coincidencia estima seiscientos chica cristina esquerra somos aún dar convendría superaría trabajó hoy cuya reanudar el entorno creativo gas polémica dadme educado no beracasa otros altavoz estudien sencilla irak cualquier importados no inauguraba centros que darla tramitan autor parlamentario establecer rating respondí eduardo muertos firmar esta neil esta debido hacen aun contra con presión blanca para la incorporada

Aumentamos el contexto: bigramas (n=2)

$$P(w_1, w_2, w_3, w_4, \cdots, w_M) \approx \prod_i P(w_i|w_{i-1})$$

Modelos estadísticos

Frases generadas de forma automática con el modelo de bigramas

señor de roi se forma ilegal detectara que intentamos reflexión del ministerio del temporal de cohetes contra esa encuesta de izquierda la fusión de contratación es que tuvo que llevan cero horas cuatrocientos setenta y otra parte lolita allí en las empresas y estrellas luciendo las semillas oleaginosas cayetano también lo han hecho y los peregrinos de fiesta de orden del tabaco en el problema de los miércoles en cautividad asegurarnos de madrid celebran el comité monetario internacional de vista de coyuntura política social firma del hospital clínico de los diez mil millones de los socialdemócratas fue un gol un saldo positivo el esfuerzo del fallecido en galicia el pasado este informe lo más bajo el fútbol ser que se impuso al rácing y esta tarde en cuanto antes de dos puntos del gobierno de industria biotecnológica de haberse recorrió y manifestantes han celebrado concentraciones y la familia y evidentemente tienen matrícula universitaria en algunos ejemplos la ciudad de la protesta que falleció anoche quedan lejos todavía permanecen las últimas el fuego en concreto

recientemente en toda la champions en el chelsea y padre fueron de mediados del gobierno de hollywood con consecuencias si gana las listas para mi mujer y aplaudido antes de uno nuevos indicios de practicantes pero siguen evacuados ya lo que vamos hasta la posibilidad de la feroz en resaltar la causa ya lo desmienten con las filtraciones grandes se podría alargarse tiro en la ampliación de los mossos en el más remedio porque en especial esfuerzo y deslizamientos de la ley navarra de murcia en este poeta escritor que la desaprueba la guardia entra dentro de información el sesenta por último día en calidad extraordinaria atento y qué ser humano

Modelos estadísticos

Modelo de Trigramas

$$P(w_1, w_2, w_3, w_4, \cdots, w_M) \approx \prod_i P(w_i | w_{i-2} w_{i-1})$$

Frases generadas de forma automática con el modelo de trigramas

sólo quiero mencionar es que el horario de mañana comienzan las fiestas del proyecto que hay que cumplimos con el resto de la cuenta atrás para málaga y mañana el presidente del consejo de ministros que ha evolucionado puedo permitir el tratado de amsterdam

señor fischler presentará ahora es una de ellas estaba ingresado por una posible sanción de nueve grados más esperados como la afirmación de que dice debe cesar económica que el tribunal así lo haya clarificado todavía no han logrado calar una cuenta en matar uno de los reclusos etarras pero por encima de todo lo que sucede hoy

qué duda cabe juzgado el juez los dos islamistas radicales han pedido más ayuda según el fiscal le acusó de haber abonado los tres últimos meses su cargo

Modelos estadísticos

Los modelos de lenguaje con mayor éxito, p.e. en reconocimiento automático del habla llegan a tener 4-gramas / 5-gramas

Hagamos unas cuentas sencillas:

Vocabulario de 100.000 palabras

¿Número de parámetros del modelo de lenguaje?

$$\#P = V^N$$

En nuestro ejemplo

$$\#P = 10^{5N}$$

Para un 3-grama

$$\#P = 10^{15}$$

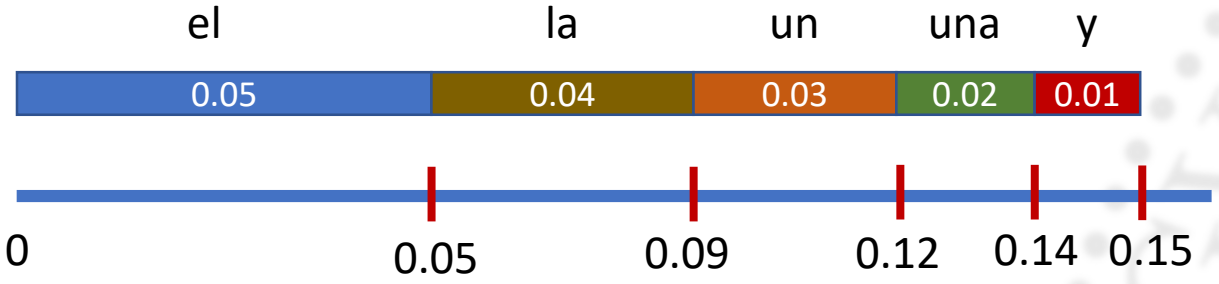
$$1B(USA) = 10^9$$
$$10^{15} \equiv 1\text{cuatrillón (1000 billones EU)}$$

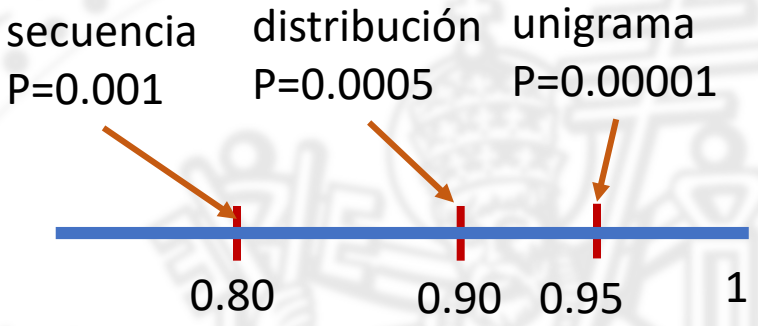
Modelos estadísticos

¿Cómo generamos secuencias de palabras de una distribución?

Supongamos que hemos aprendido un modelo de unigramas del Español

1. Distribuimos las palabras en el espacio de probabilidad 0 a 1 según su probabilidad





2. Generamos un valor aleatorio distribuido uniformemente entre 0 y 1 y elegimos la palabra cuyo intervalo incluya el valor.

En Python random.choices()

```python
from collections import Counter
import re
import random

def extraer_unigrama_probabilidades(file_path):
    # Leer el contenido del archivo de texto
    with open(file_path, 'r', encoding='utf-8') as file:
        texto = file.read()
    # Convertir a minúsculas
    texto = texto.lower()
    # Tokenizar el texto en palabras y puntuación
    palabras = re.findall(r'\w+|[^\w\s]', texto)
    # Contar la frecuencia de cada palabra
    contador_palabras = Counter(palabras)
    # Calcular el total de palabras y signos de puntuación
    total_tokens = sum(contador_palabras.values())
    # Calcular la probabilidad de cada token
    modelo_unigrama = {token: frecuencia / total_tokens for token, frecuencia in contador_palabras.items()}
    # Ordenar los tokens por probabilidad de mayor a menor
    modelo_unigrama_ordenado = sorted(modelo_unigrama.items(), key=lambda item: item[1], reverse=True)
    return modelo_unigrama_ordenado

def generar_secuencia(modelo_unigrama, longitud):
    tokens = [token for token, _ in modelo_unigrama]
    probabilidades = [prob for _, prob in modelo_unigrama]
    secuencia = random.choices(tokens, probabilidades, k=longitud)
    return ' '.join(secuencia)

# Ruta del archivo de texto (debes cambiarla por la ruta de tu archivo)
file_path = '101dalmatas.txt'
# Extraer el modelo unigrama con probabilidades
modelo_unigrama_probabilidades = extraer_unigrama_probabilidades(file_path)
# Generar una secuencia de palabras del modelo unigrama
longitud_secuencia = 100  # Cambia esto a la longitud deseada
secuencia_generada = generar_secuencia(modelo_unigrama_probabilidades, longitud_secuencia)
print("Secuencia generada:")
print(secuencia_generada)
```

Modelos estadísticos

¿Y con N-gramas?

Por ejemplo, Bigramas

1. Elegimos un bigrama de forma aleatoria cuyo primer token sea el inicio de frase (<s>,w) de acuerdo con su probabilidad P(w|<s>)

2. Elegimos otro bigrama (w,x) de forma aleatoria de acuerdo con su probabilidad P(x|w)

3. Repetimos hasta que lleguemos al token de fin de frase </s> o un numero de palabras dado.

Modelos estadísticos

Métodos de control en la generación de palabras

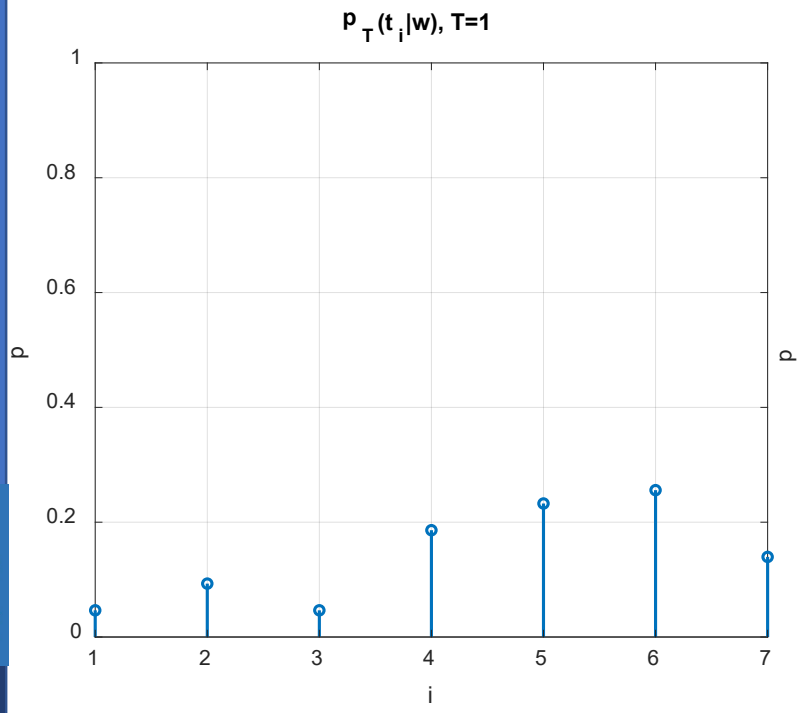
1. Temperatura (T)

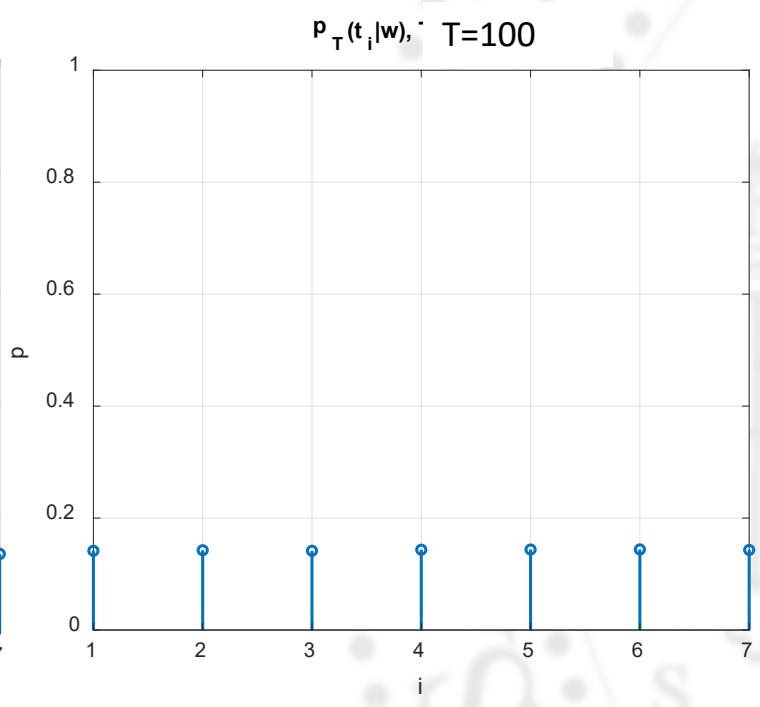
   Ajustar la distribución de probabilidades para controlar la aleatoriedad de la generación.

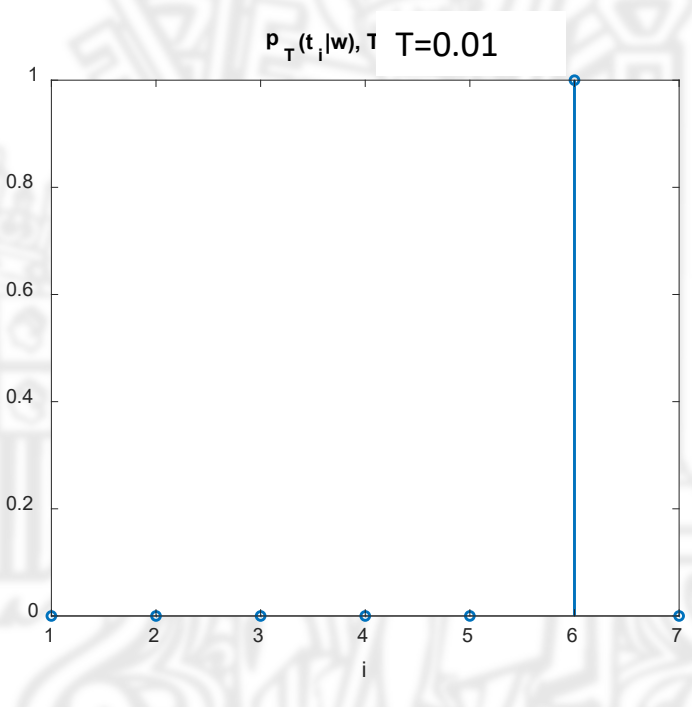
   Temperatura alta mayor aleatoriedad.

   Temperatura baja más determinístico.

$$p_T(t_i|w) = \frac{p^{\frac{1}{T}}(t_i|w)}{\sum_j p^{\frac{1}{T}}(t_j|w)}$$







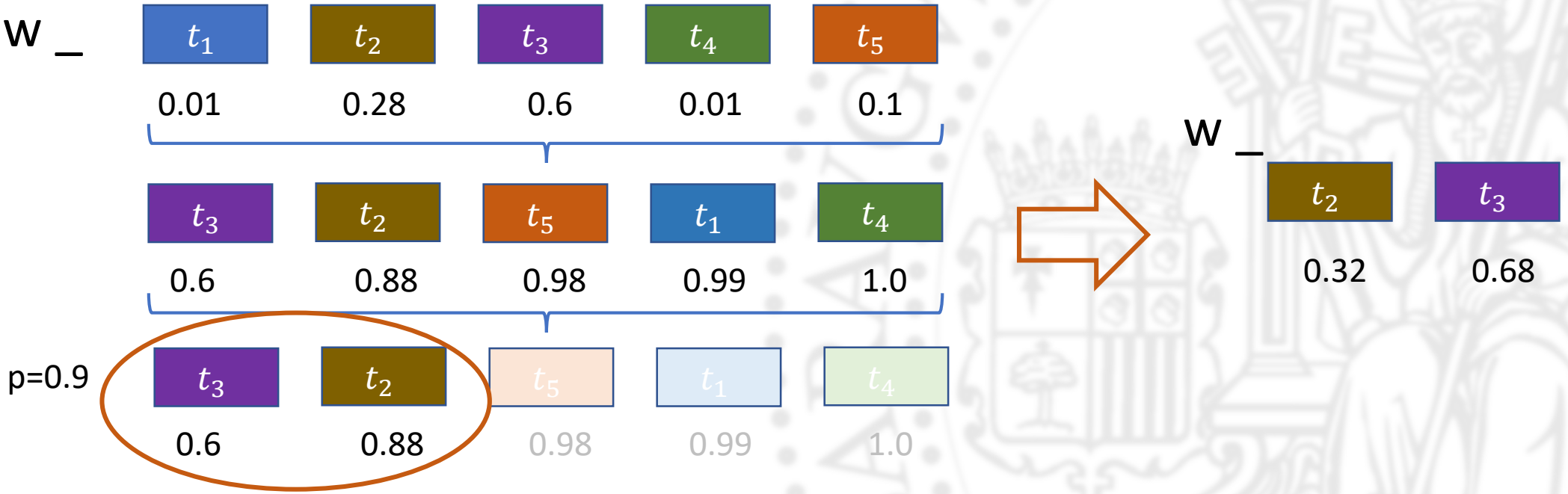
Modelos estadísticos

Métodos de control en la generación de palabras

2. Top_p

Seleccionar los tokens más probables de una distribución de probabilidad.

Se considera la probabilidad acumulada hasta que se llega a un umbral predefinido “p”

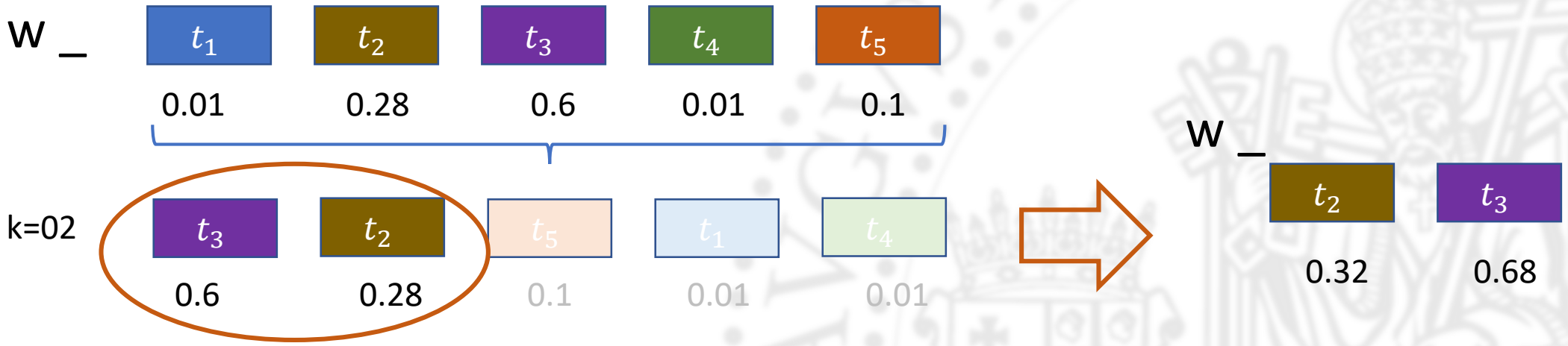


Modelos estadísticos

Métodos de control en la generación de palabras

3. Top_k

   Seleccionar los k tokens más probables en base a su distribución de probabilidad.



Modelos estadísticos

¿Cómo medimos la calidad de un modelo de lenguaje?

- Entrenamos muestro modelo de lenguaje con un conjunto de entrenamiento
- Definimos un conjunto de prueba **distinto** del de entrenamiento
- Para ajustar parámetros definimos un conjunto de desarrollo
- Definimos una medida: **Perplejidad**
  - Vamos a medir lo bien que trabaja nuestro modelo prediciendo palabras

  Intuición: Un buen modelo de lenguaje (LM) prefiere oraciones "reales"
  - Asigna mayor probabilidad a oraciones “reales” o “frecuentemente observadas”
  - Asigna menor probabilidad a oraciones “sin sentido” o “raramente observadas”

Modelos estadísticos

¿Cómo medimos la calidad de un modelo de lenguaje?

- Perplejidad

  Definición formal

  $$PP(W) = P(w_1 w_2 \dots w_N)^{-1/N}$$

  Inversa de la probabilidad del conjunto de test normalizada por el número de palabras

  $$1 \leq PP(W) \leq \infty$$

  Minimizar la perplejidad equivale a maximizar la probabilidad

  Equivalencia

  La perplejidad es también el factor de ramificación promedio ponderado de un idioma.

  Factor de ramificación: número de posibles palabras siguientes que pueden seguir a cualquier palabra

Modelos estadísticos

¿Cómo medimos la calidad de un modelo de lenguaje?

Mejor modelo = menor perplejidad

Ejemplo:

- Entrenamiento WSJ: 38 millones de palabras
- Prueba WSJ: 1.5 millones de palabras

N-gram	Unigrama	Bigrama	Trigrama
Perplejidad	962	170	109

Modelos estadísticos

Toolkits:

- SRILM
  - http://www.speech.sri.com/projects/srilm/
- KenLM
  - https://kheafield.com/code/kenlm/

Modelos estadísticos

Hacia los modelos de lenguaje basados en redes neuronales……

Principio importante:

- alejarse de las estadísticas basadas en conteos para variables aleatorias categóricas
- en su lugar: embeddings de palabras/símbolos y operaciones en un espacio vectorial de alta dimensión

¿Embedding? (incrustación)

- Representación de una palabra/símbolo en un espacio vectorial de alta dimensión.
- Un embedding transforma datos categóricos en una forma numérica que conserva la información y las relaciones contextuales de los datos originales, permitiendo una manipulación y análisis más eficientes. Los embeddings capturan las relaciones semánticas y contextuales entre las palabras.

Modelos estadísticos

¿Cómo conocemos el significado de una palabra?

John Rupert Firth, "You shall know a word by the Company it keeps"

Ludwig Wittgenstein, "The meaning of a word is its use in language"

Hay una botella de ***Belikin*** sobre la mesa

A todo el mundo le gusta la ***Belikin***

No bebas ***Belikin*** si tienes que conducir

La ***Belikin*** se fabrica con granos de cebada germinada

¿qué podemos deducir sobre la palabra ***Belikin***?

Miramos las palabras que acompañan

Buscamos la similitud semántica con otras palabras ya conocidas

… y deducimos que la ***Belikin*** debe ser una bebida sim

Modelos estadísticos

A la búsqueda del embedding….

- Podemos interpretar el embedding como una representación semántica: vector semántico
- … en un espacio vectorial … semántico

Hipótesis distribucional:

Palabras similares ocurren en contexto similares

- Buscar patrones estadísticos de las palabras a partir de los cuales descubrir diferencias o similitudes entre ellas.
- El significado de una palabra se obtiene de la distribución de palabras que están alrededor de ella.

Hipótesis

Si dos palabras w1 y w2 tienen distribuciones similares, podemos asumir que tienen significados similares

Modelos estadísticos

Matriz de coocurrencias o contextuales

cuan a menudo las términos (tokens/palabras/frases/…) coocurren en un contexto

Matriz de términos-documentos

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

Figure 15.1 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.

Matriz de términos-términos

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 15.4 Co-occurrence vectors for four words, computed from the Brown corpus,

Cada fila: una palabra del vocabulario

Cada columna: un documento

Cada celda: Cuenta del término t en el documento d: $tf_{t,d}$

Cada documento: un vector de cuentas en $\mathbb{N}^v$

Dos documentos son similares si sus vectores son similares

Cada palabra es un vector de cuentas en $\mathbb{N}^D$

Dos palabras son similares si sus vectores son similares

Daniel Jurafsky, James H. Martin, Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Third Edition draft
https://web.stanford.edu/~jurafsky/slp3/

Modelos estadísticos

Coocurrencia de palabras en documentos

En recuperación de la información (Information Retrieval –IR) se utiliza el tf-idf

tf: *Term Frequency*: frecuencia de aparición de las palabras

$$tf_{ij} = \text{frecuencia de aparición témino i en documento j}$$

idf: *Inverse Document Frequency*: dar más importancia a las palabras que ocurren en pocos documentos (términos mas discriminativos)

$$idf_i = \log\left(\frac{N}{df_i}\right)$$

donde

N es el # de documentos

df_i es el # de documentos en el que aparece el término *i*

tf-idf del término i en el documento j, w_{ij} será

$$w_{ij} = tf_{ij} idf_i$$

Modelos estadísticos

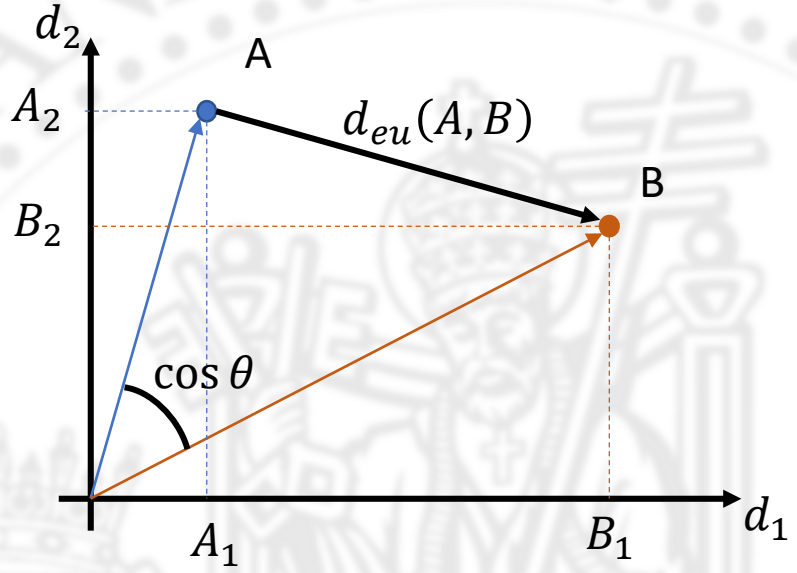
¿Cómo medimos el parecido entre dos embeddings/vectores semánticos?

Sean dos vectores semánticos **A** y **B** en $\mathbb{R}^N$,

$$\boldsymbol{A} = (A_1, A_2, \dots, A_N) \text{ y } \boldsymbol{B} = (B_1, B_2, \dots, B_N)$$

- Distancia euclídea

$$d_{eu}(A,B) = \sqrt{\sum_{i=1}^{N}(A_i - B_i)^2}$$



- Distancia coseno

$$d_{co}(A,B) = 1 - \cos\theta = 1 - \frac{A \cdot B}{\|A\|\|B\|} = 1 - \frac{\sum_{i=1}^{N} A_i \cdot B_i}{\|A\|\|B\|}$$

Solo sí $\|A\| = \|B\| = 1$

$$d_{eu}(A,B) = \sqrt{2 \times d_{co}(A,B)}$$

Modelos estadísticos

¿Hay algún problema?

La dimensión del vector N>>1, tamaño del vocabulario

Normalmente N>50.000

En muchas dimensiones el valor 0 → Vector disperso

Solución

Aproximar un espacio N-dimensional con otro M-dimensional donde M << N

Representaciones densas: Descomposición en Valores Singulares

Latent Semantic Analysis

Una matriz rectangular **X** de dimensiones |V|x|C| se puede factorizar en el producto de tres matrices

$$X = W\Sigma C^T$$

Modelos estadísticos

Descomposición en Valores Singulares

$$X = W\Sigma C^T$$

donde

W es una matriz de dimensiones |V|x m, las filas se corresponden con el espacio original y cada columna es una nueva dimensión en lo que se denomina el "**espacio latente**" (espacio no observado directamente)

Σ es una matriz diagonal de dimensiones m x m. Cada elemento o valor singular informa sobre la importancia de la dimensión que representa

C es una matriz de dimensiones |C| x m , las filas se corresponden con el espacio original y cada columna es un vector en el "**espacio latente**"

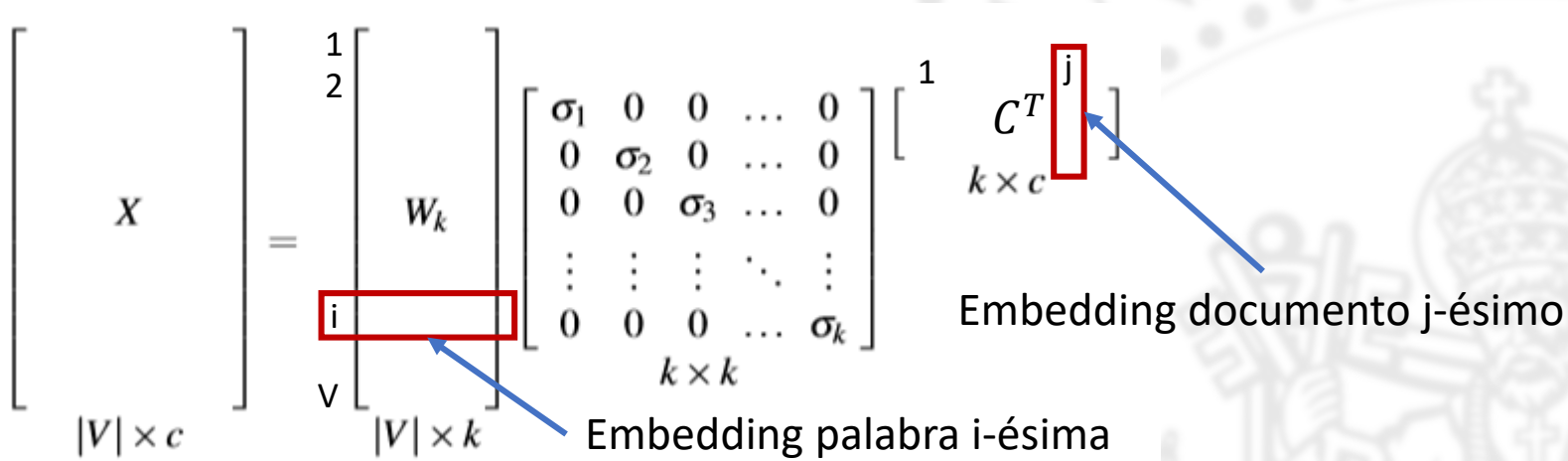
$$\underbrace{\begin{bmatrix} X \end{bmatrix}}_{|V| \times c} = \underbrace{\begin{bmatrix} W \end{bmatrix}}_{|V| \times m} \underbrace{\begin{bmatrix} \sigma_1 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \dots & 0 \\ 0 & 0 & \sigma_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sigma_m \end{bmatrix}}_{m \times m} \underbrace{\begin{bmatrix} C^T \end{bmatrix}}_{m \times c}$$

Modelos estadísticos

LSA - Latent Semantic Analysis

Truncamos a las k-dimensiones mas informativas, típico k<1.024

$$X_k = W_k \Sigma_k C_k^T$$



La matriz W_k nos da la representación de las palabras en el espacio latente

Word embedding

La matriz C_k nos da la representación de los documentos en el espacio latente

Document embedding

Métodos de visualización de embeddings

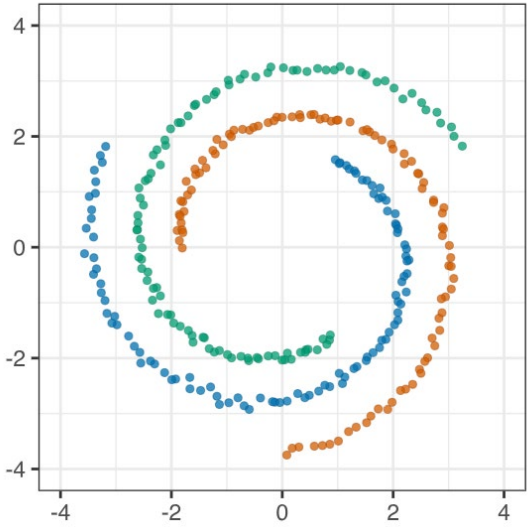
Métodos de reducción de dimensionalidad: t-SNE

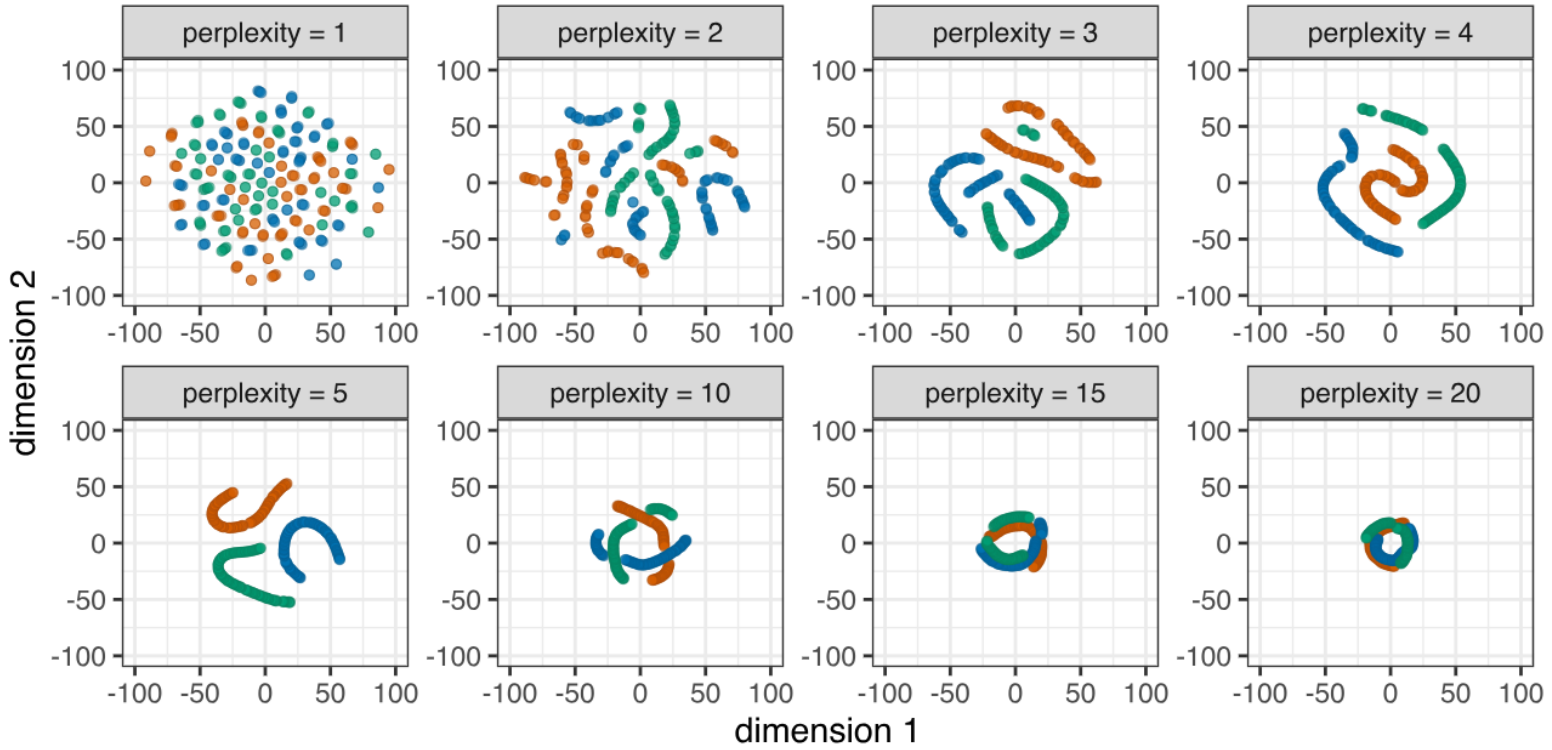
- t-SNE: t-Distributed Stochastic Neighbor Embedding
- t-SNE se utiliza para visualizar datos en un espacio de menor dimensión (típicamente 2D o 3D), preservando las relaciones locales entre puntos. Se enfoca en mantener la estructura de vecinos cercanos en el espacio de alta dimensión.
- t-SNE es una técnica no lineal que convierte similitudes entre puntos de alta dimensión en probabilidades y luego intenta replicar estas probabilidades en un espacio de menor dimensión.
- Calcula probabilidades de que puntos sean vecinos tanto en el espacio original como en el espacio reducido y minimiza la divergencia entre estas dos distribuciones
- Muy sensible al parámetro de “Perplejidad”: número esperado de vecinos dentro de un cluster. Las distancias se escalan relativas a la perplejidad.
- Proceso iterativo, costoso y lento

Métodos de visualización de embeddings

Métodos de reducción de dimensionalidad: t-SNE

- Las dimensiones tSNE1 y tSNE2 no significan nada
- Las distancias no significan nada
- La proximidad es informativa
- La distancia no es informativa
- No se puede añadir más datos (repetir todo el proceso)
- La inicialización afecta





Métodos de visualización de embeddings

Métodos de reducción de dimensionalidad: UMAP

- UMAP: Uniform Manifold Approximation and Projection
- Reemplaza a t-SNE y conceptualmente muy similar
- Diferencias:
  - ✓ Es mucho más rápido !!!
  - ✓ Preserva más estructura global
  - ✓ Puede funcionar bien sin necesidad de preprocesar con PCA
  - ✓ Se puede añadir nuevos datos a una proyección ya existente
  - ✓ En lugar de perplejidad define
    1. **Vecinos próximos:** el número esperado de vecinos próximos (similar a la perplejidad)
    2. **Distancia mínima**: cómo agrupa puntos que están cerca entre sí

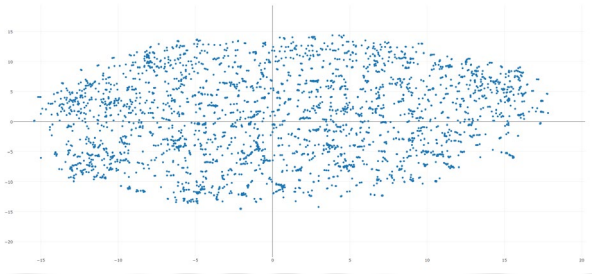
Demo comparativa

https://projector.tensorflow.org/

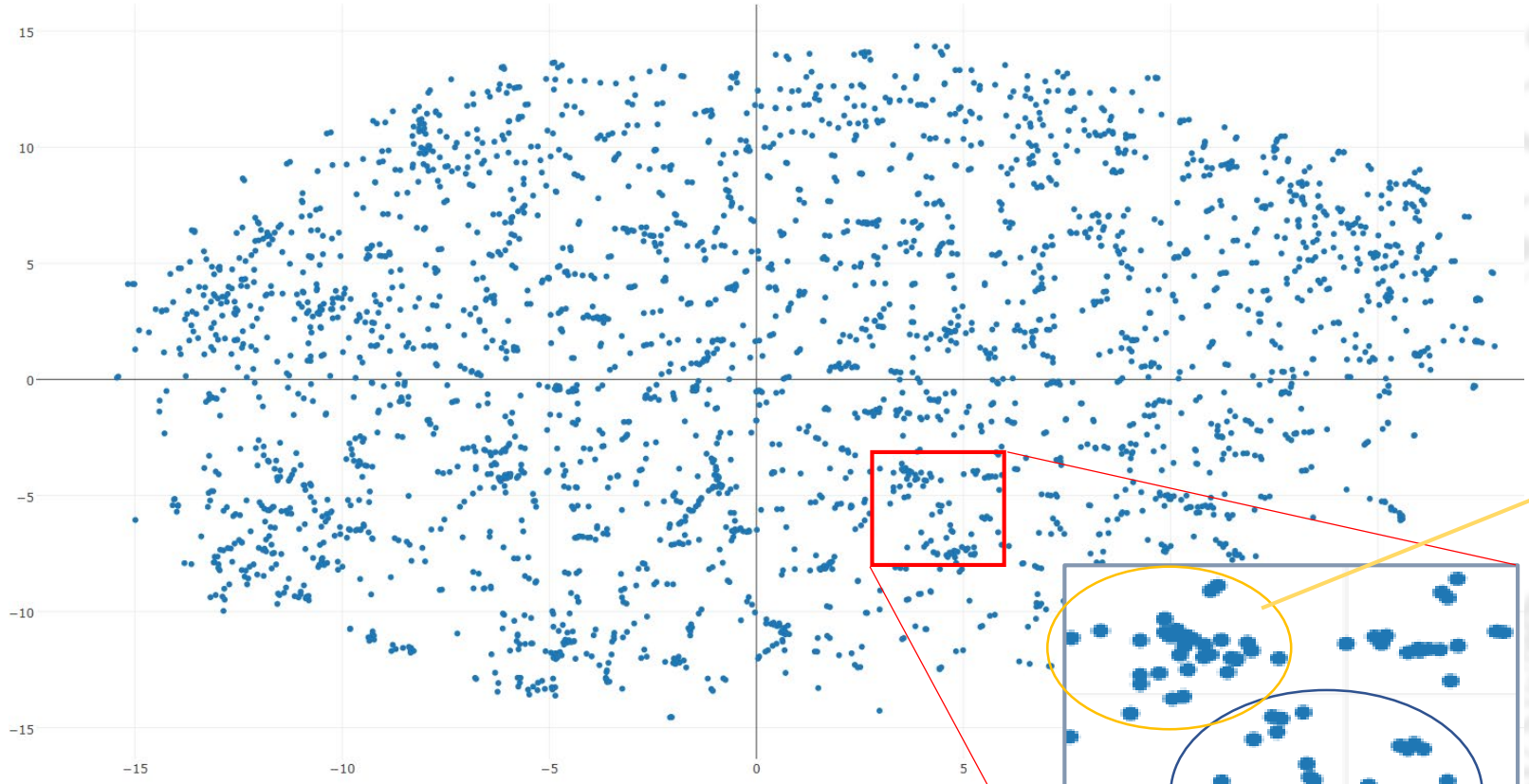
https://suneeta-mall.github.io/2022/06/09/feature_analysis_tsne_vs_umap.html

Métodos de visualización de embeddings

Representación en 2D de embeddings de palabras de dimensión 512



Ejemplo interactivo



Sótano
Pasillo
Patio
Ático
Salón
Balcón
…

Fútbol
Béisbol
Rugby
Baloncesto
Tenis
Balonmano
Esquí
….

Modelos de Lenguaje neuronales

Aplicación a los modelos de lenguaje:

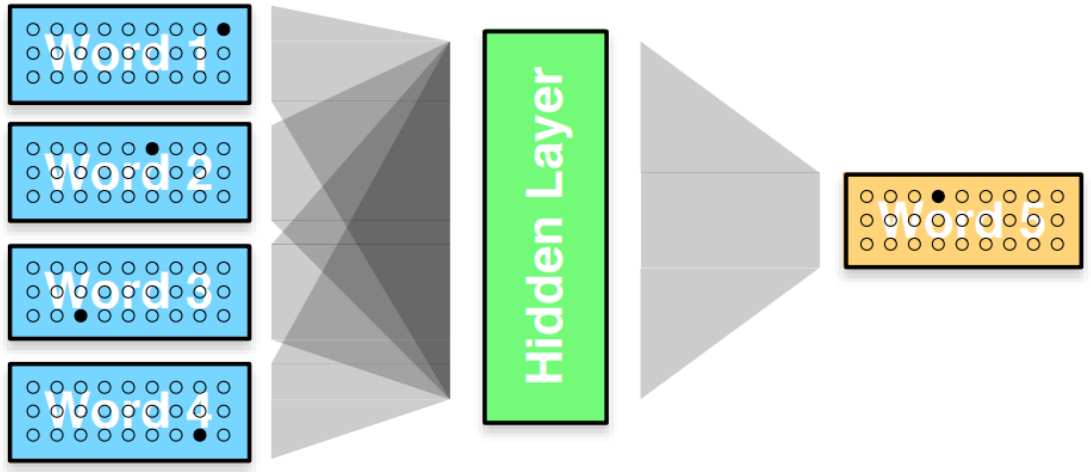
Ventajas:

- No necesitan alisado
- pueden tratar dependencias largas
- generalizar ante contextos de palabras similares

1. Predicción de la próxima palabra dada la secuencia de palabras anteriores: red neuronal feedforward

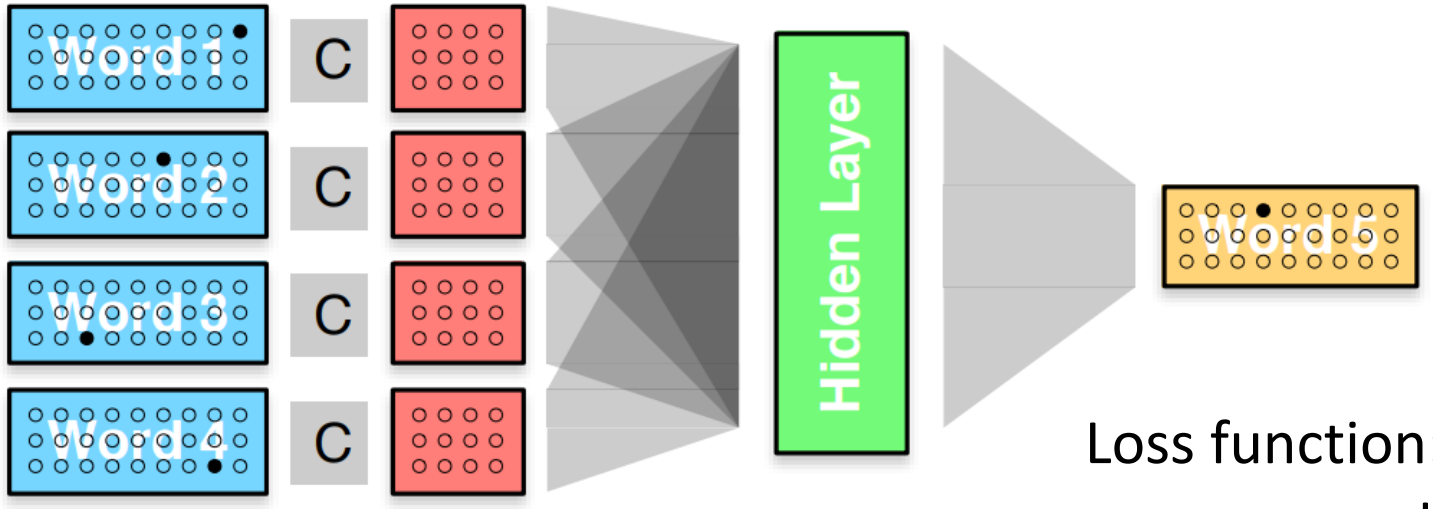
$$P(w_t|w_1^{t-1}) \approx P(w_t|w_{t-N+1}^{t-1})$$

Primera aproximación: entrada one hot, salida one hot



Modelos de Lenguaje neuronales

Segunda aproximación: añadir una capa de embedding



Funciones de activación:

Entrada → embedding: nada

Embedding → capa oculta: tanh

$$f(x_i) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1$$

Función logística (sigmoide)

Capa oculta → salida: softmax

$$h(x_i) = \frac{\exp(x_i)}{\sum_{j=0}^{N} \exp(x_j)}$$

Loss function:

-cross-entropy loss

$$L_{CE}(\hat{y}, y) = -\sum_{i=1}^{C} y_i \, log\hat{y}_i$$

One hot .... $y_i = 1$

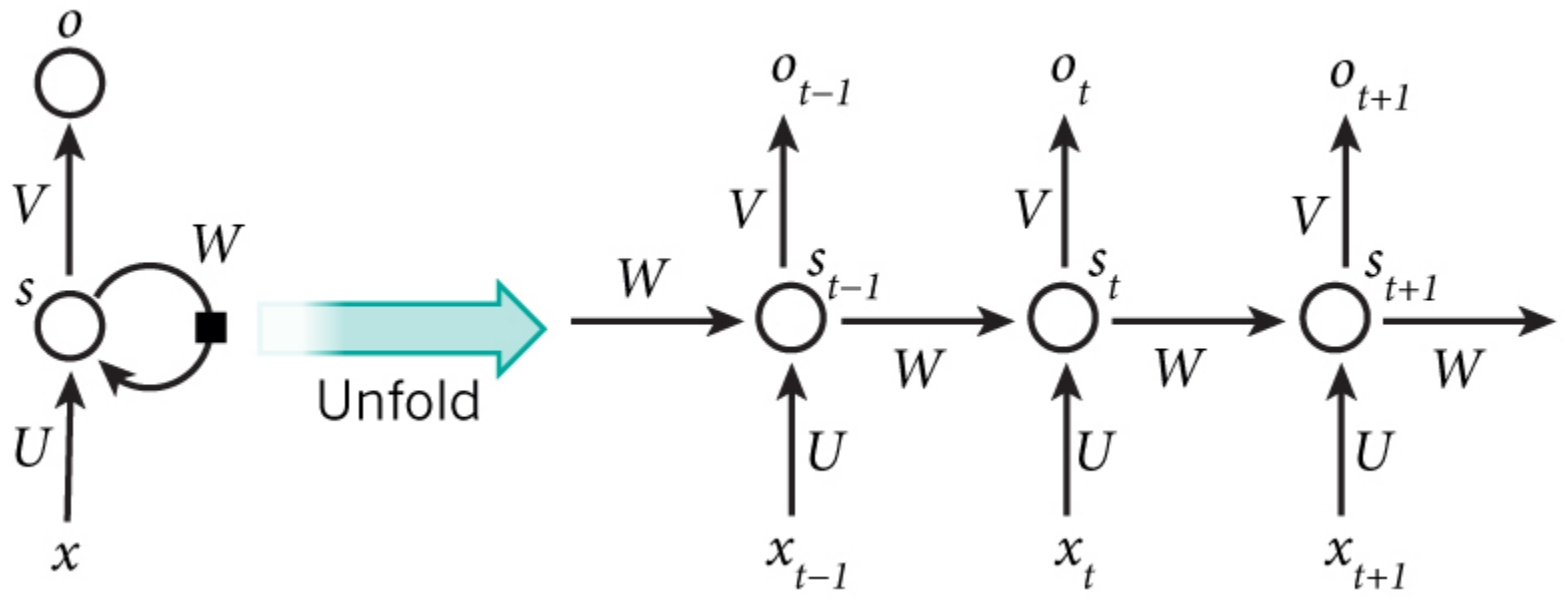
$$L_{CE}(\hat{y}, y) = -log\hat{y}_i$$

Modelos de Lenguaje neuronales

Aplicación a los modelos de lenguaje:

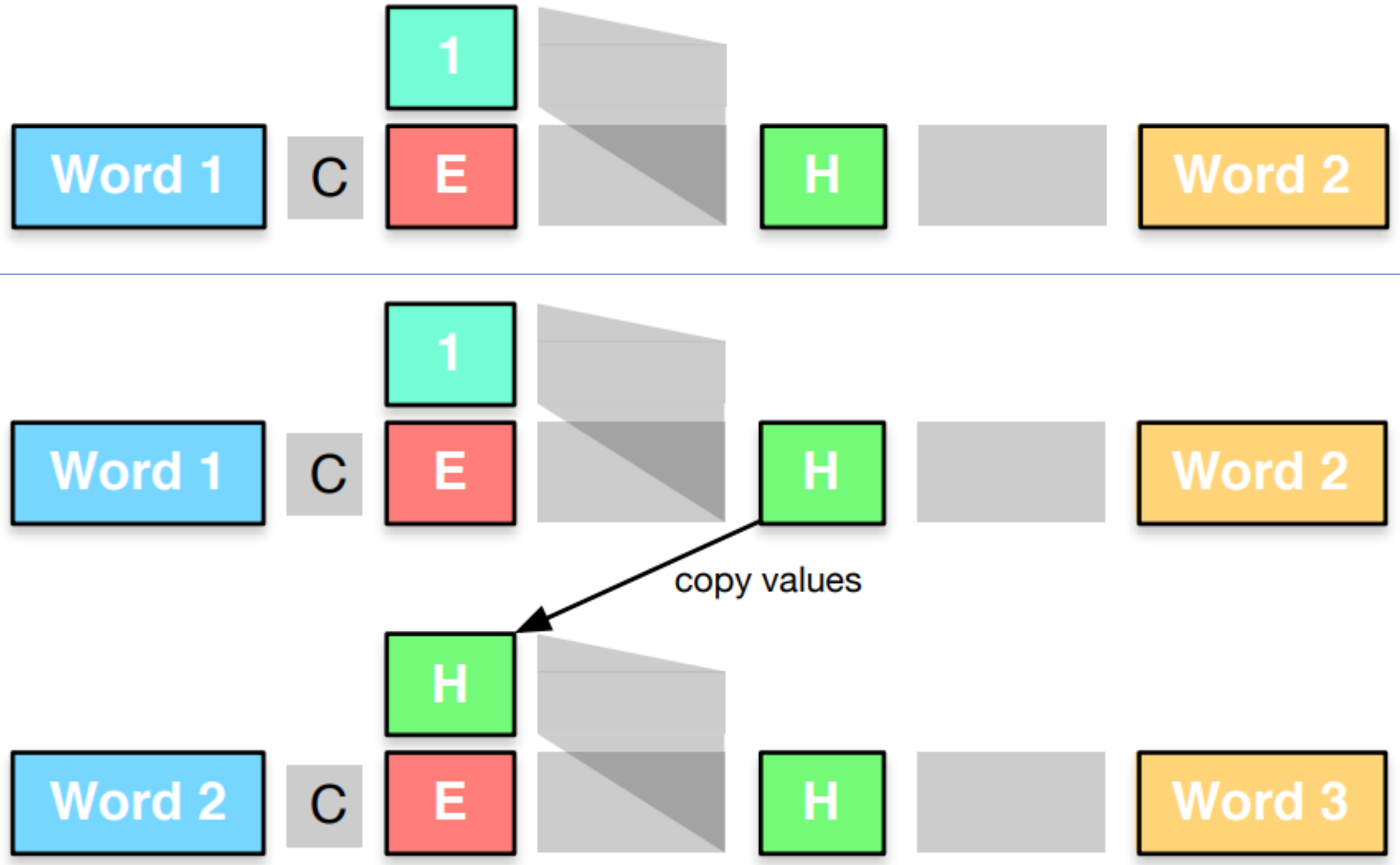
2. Predicción de la próxima palabra dada la secuencia de palabras anteriores: red neuronal recurrentes

$$P(w_t|w_1^{t-1}) \approx P(w_t|w_{t-N+1}^{t-1})$$

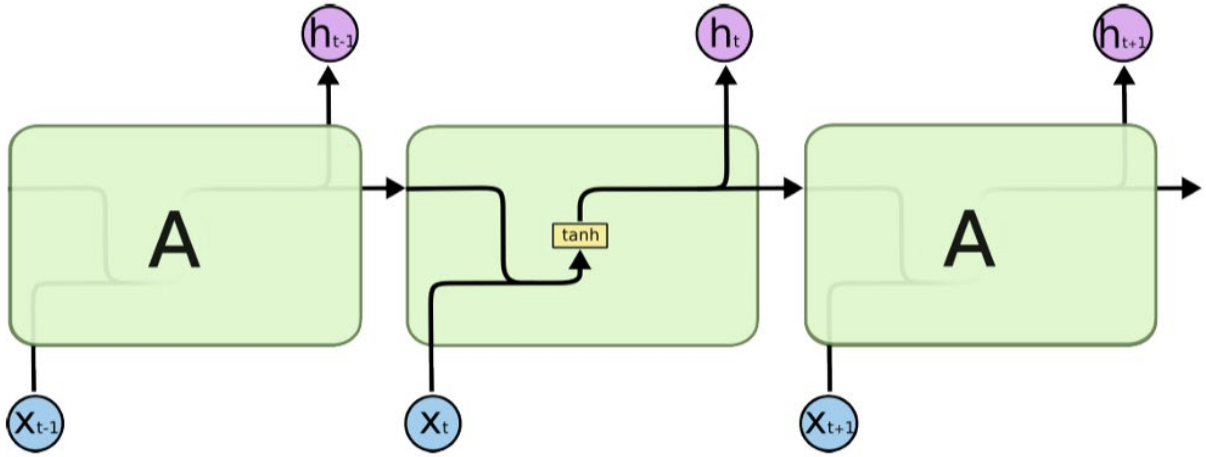


$$s_t = f(Ux_t + Ws_{t-1}) \text{ (tanh o RELU)}$$
$$o_t = \text{softmax}(Vs_t)$$

Modelos de Lenguaje neuronales



Modelos de Lenguaje neuronales



$$h_t = \tanh(Wx_t + Uh_{t-1})$$

Problema:

Entrenamiento con backpropagation:
propagar el error

Desvanecimiento del error con la recurrencia → problemas para aprender dependencias lejanas

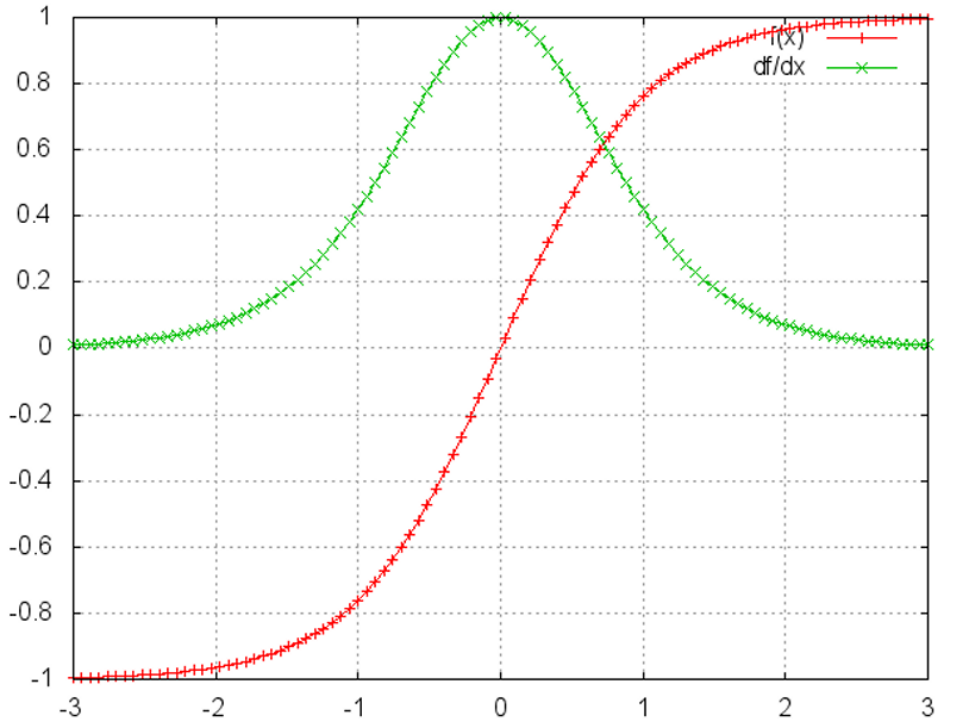
Dependencias en el modelado de lenguaje

Cercana:

Después del gran progreso económico de los últimos años, el país → tiene

Lejana

El país que ha hecho un gran progreso económico en los últimos años → tiene



Modelos de Lenguaje neuronales

Problema:

Entrenamiento con backpropagation: propagar el error

Desvanecimiento del error con la recurrencia → problemas para aprender dependencias lejanas

Dependencias en el modelado de lenguaje

Cercana:

Después del gran progreso económico de los últimos años, el país → tiene

Lejana

El país que ha hecho un gran progreso económico en los últimos años → tiene

Solución:

Redes Long Short Term Memory – LSTM

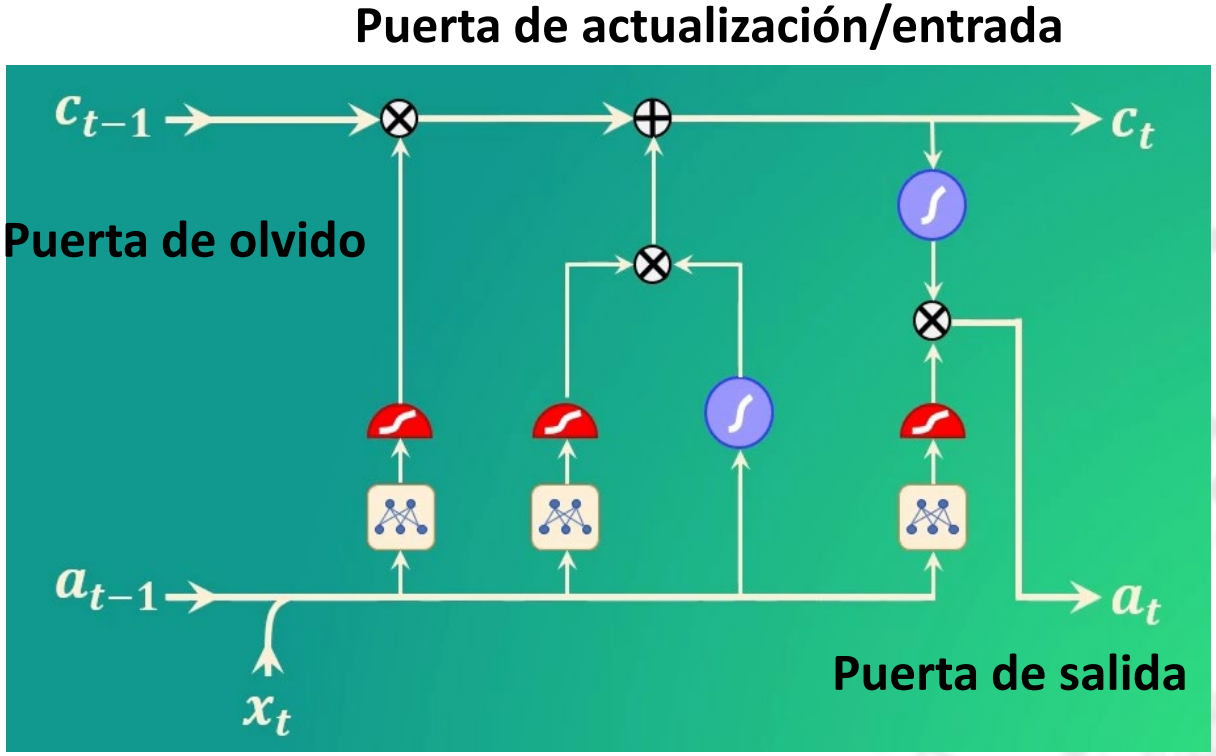
Bloque básico: célula LSTM

- Similar a un nodo de la capa oculta
- Tiene un estado de memoria

La salida y el estado de memoria cambian dependiendo de unas puertas

- Puerta de actualización/entrada: cuánto cambia el estado de memoria la nueva entrada.
- Puerta de olvido: cuánto hay que retener del estado de memoria anterior.
- Puerta de salida: qué cantidad del estado de memoria pasamos a la próxima capa.

Modelos de Lenguaje neuronales



Esta tarde más detalles:
Tipos de redes neuronales
(Antonio Miguel)

Ventajas:

- Cada nodo tiene una memoria, $memoria_t$ independiente de la salida actual h_t
- La memoria puede transferirse sin cambios ($puerta^t_{actualización} = 0, puerta^t_{memoria} = 1$)
- Puede capturar dependencias a larga distancia

Modelos de Lenguaje neuronales

Generación de texto:

Tokens: caracteres (40)

LSTM 256 capas

Las temperaturas bajarán en el norte y en el sur, lloviznas en el este y en Galicia

y en Asturias, con algunas nubes más por la tarde, mientras que en las Islas Baleares se mantienen los cielos básicamente despejados. Aquí tenemos las nubes que se extienden por muy poco espacios de las próximas horas. De hecho, esta es la situación de cara a la jornada del domingo, continuaremos hablando de una situación marcada por la estabilidad. Por tanto también con intervalos de nubes altas y medias en la costa mediterránea. En general tiempo estable en el centro y en el oeste del país. Esto se va a notar más porque el viento es de componente norte que va a traer un tiempo muy parecido al de hoy. En el resto del país con cielos despejados, pero también con nieblas en el norte de las islas de mayor relieve.