Procesamiento de Lenguaje Natural

Modelos de Lenguaje

Mauricio Toledo-Acosta
mauricio.toledo@unison.mx

Departamento de Matemáticas
Universidad de Sonora

Section 1

Introducción

Referencias

- **Chapter 3**. Jurafsky, D., Martin, J. H. (2019). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.
- **Chapter II.6**. Eisenstein, J. (2018). Natural language processing. Jacob Eisenstein.

Objetivo

¿Cuál es la siguiente palabra?

En el parque, los niños juegan con ...

Objetivo

¿Cuál es la siguiente palabra?

En el parque, los niños juegan con ...

¿Cuál es la siguiente palabra?

El agua hierve a 100 grados ...

Modelos de lenguaje

Los **modelos de lenguaje** son modelos que asignan una probabilidad a secuencias de palabras.

¿Cuál es más probable?

- *café el en mesa la sobre libro un dejé olvidé y*

Modelos de lenguaje

Los **modelos de lenguaje** son modelos que asignan una probabilidad a secuencias de palabras.

¿Cuál es más probable?

- *café el en mesa la sobre libro un dejé olvidé y*
- *dejé un libro sobre la mesa en el café y lo olvidé*

¿Por qué nos interesan estas probabilidades?

Las probabilidades son esenciales en cualquier tarea en la que tengamos que identificar palabras en entradas ruidosas y ambiguas, como el reconocimiento de voz o de escritura.

- **Traducción automática**:

$$P(\text{high winds tonight}) > P(\text{large winds tonight})$$

- **Corrección ortográfica**: *The office is about fileen minuets from my house*.

$$P(\text{about fifteen minutes from}) > P(\text{about fifteen minuets from})$$

¿Por qué nos interesan estas probabilidades?

- **Reconocimiento del habla**

$$P(\text{I saw a van}) > P(\text{Eyes awe of an})$$

- **Respuesta de preguntas**
- **Generación de texto**

Tipos de Modelos de Lenguaje

Hay dos métodos para construir modelos de lenguaje:

1. **Modelos de Lenguaje Estadísticos**. Predicen la siguiente palabra dadas las palabras que le preceden, esto lo hacen usando conteos de co-ocurrencias. El modelo de *n*-gramas es el más sencillo.
2. **Modelos de Lenguaje Neuronales**. Predicen la siguiente palabra usando embeddings que capturan diversos fenómenos lingüísticos a partir del uso de redes neuronales.
   - Modelos neuronales *clásicos*: `Word2Vec`, `FastText`, `GloVe`, `ConceptNet`, ...
   - Modelos basados en mecanismos de atención: `Bert`, `LlaMa`, `GPT`, `Claude`, `ModernBert`, `DeepSeek`, ...

¿Cómo se calculan las probabilidades?

Queremos calcular la probabilidad de una secuencia de palabras $W = w_1, w_2, ..., w_n$. Esto lo hacemos con la probabilidad conjunta:

$$P(W) = P(w_1, w_2, ..., w_n).$$

Una tarea relacionada es calcular la probabilidad condicional

$$P(w_n \mid w_1, w_2, ..., w_{n-1})$$

Ambas estan relacionadas por la regla de la cadena

$$P(W) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \cdots P(w_n \mid w_1, w_2, ..., w_{n-1})$$

¿Cómo se calculan las probabilidades?

its water is so transparent that the

Podemos estimar probabilidades en términos de conteos:

$$P(\text{ the } | \text{ its water is so transparent that }) = \frac{\text{contar}(\text{ its water is so transparent that the })}{\text{contar}(\text{ its water is so transparent that })}$$

Esto se llama *frecuencia relativa*.

Simplificando con la suposición de Markov

La distribución de probabilidad del valor futuro de una variable aleatoria depende únicamente de su valor presente, siendo independiente de la historia de dicha variable.

Podemos hacer las siguientes aproximaciones:

$$P(w_n|w_1 w_2 \ldots w_{n-1}) \approx P(w_n)$$
$$P(w_n|w_1 w_2 \ldots w_{n-1}) \approx P(w_n|w_{n-1})$$
$$P(w_n|w_1 w_2 \ldots w_{n-1}) \approx P(w_n|w_{n-2} w_{n-1})$$
$$\ldots$$

Section 2

Modelo de *n*-gramas

Modelo de *n*-gramas

Please turn your homework tomorrow

Un ***n*-grama** es una secuencia de N palabras: un 2-grama (o bigrama) es una secuencia de dos palabras como *please-turn*, *turn-your* *your-homework*, y un 3-grama (o trigrama) es una secuencia de tres palabras como *please-turn-your* o *turn-your-homework*.

El **modelo de *n*-gramas** estima la probabilidad de una palabra dada una secuencia de palabras.

Al decir *n*-gramas nos referimos a las secuencias o al modelo predictivo que asigna probabilidades.

Usos del modelo de *n*-gramas

Hay dos principales utilidades de este modelo:

1. **Tareas secuenciales**: Modelar la dependencia entre elementos consecutivos en una secuencia (palabras, caracteres, etc.). Por ejemplo: Generación de texto, corrección ortográfica, etc.
2. **Tareas no secuenciales**: Representar el contenido sin considerar el orden estricto, enfocándose en la presencia o frecuencia de patrones. Ejemplos: Tareas de clasificación, recuperación de información, etc.

Ejemplo

Hoy me gusta comer tacos. Me gusta mucho comer tacos hoy. También me gusta comer tamales, pero hoy prefiero tacos. Me gusta mucho.

	me	gusta	comer	tacos	tamales	mucho	hoy
me	5	50	0	2	1	0	3
gusta	2	0	40	1	4	6	0
comer	0	2	0	30	25	0	0
tacos	1	0	3	0	5	2	1
tamales	0	1	2	5	0	1	0
mucho	4	6	0	0	1	0	8
hoy	2	0	1	1	0	7	0

Cálculo de probabilidades condicionales para 2-gramas

El cálculo de probabilidades en modelos basados en 2-gramas se realiza mediante la siguiente fórmula:

$$P(w_i|w_{i-1}) = \frac{\text{Frecuencia}(w_{i-1}, w_i)}{\sum_j \text{Frecuencia}(w_{i-1}, w_j)}$$

- $\text{Frecuencia}(w_{i-1}, w_i)$: Número de veces que la palabra w_i aparece inmediatamente después de w_{i-1} en el corpus.
- $\sum_j \text{Frecuencia}(w_{i-1}, w_j)$: Suma de todas las frecuencias de palabras que siguen a w_{i-1}.

Ejemplo: Si en el corpus tenemos:

- $\text{Frecuencia}(\text{"me"}, \text{"gusta"}) = 50$,
- Frecuencia total para "me" $= 61$,

$$P(\text{gusta}|\text{me}) = \frac{50}{61} \approx 0.82$$

Ejemplo: Probabilidades

	me	gusta	comer	tacos	tamales	mucho	hoy
me	0.082	0.820	0.000	0.033	0.016	0.000	0.049
gusta	0.033	0.000	0.667	0.017	0.067	0.100	0.000
comer	0.000	0.032	0.000	0.484	0.403	0.000	0.000
tacos	0.020	0.000	0.060	0.000	0.100	0.020	0.020
tamales	0.000	0.029	0.086	0.200	0.000	0.029	0.000
mucho	0.211	0.316	0.000	0.000	0.053	0.000	0.421
hoy	0.167	0.000	0.083	0.083	0.000	0.667	0.000

Observa las sumatorias por fila.

¿Para qué calcular estas probabilidades?

Las probabilidades condicionales de n-gramas se utilizan en modelos de lenguaje para:

- Generar texto: Predecir la siguiente palabra en una secuencia, basándose en el contexto inmediato.
- Corrección gramatical: Identificar combinaciones de palabras poco probables.
- Traducción automática: Evaluar la fluidez de frases traducidas.

Limitaciones:

- Los n-gramas solo capturan dependencias locales.
- Requiere grandes corpus para evitar probabilidades cero en combinaciones raras.
- No entiende significado, solo patrones estadísticos.

Algunas ventajas

- Los *n*-gramas ayudan a capturar la información contextual y la semántica dentro de una secuencia de palabras, proporcionando una comprensión más matizada del lenguaje.
- En tareas de recuperación de información (information retrieval), los *n*-gramas ayudan a emparejar y clasificar documentos según la relevancia de los patrones de *n*-gramas.
- Los *n*-gramas sirven como características poderosas en la clasificación de texto y el análisis de sentimientos, capturando patrones significativos que contribuyen a la caracterización de diferentes clases o sentimientos.

Algunas desventajas

- En general es un modelo de lenguaje insuficiente. Por ejemplo, no captura dependencias lejanas

> The computer which I had just put into the machine room on the fifth floor crashed.

  En el caso anterior el bigrama *computer-crashed* puede ser un bigrama importante.
- El lenguaje es creativo, todo el tiempo se crean asociaciones nuevas, y no siempre podremos contar frases enteras.
- Problemas de escalabilidad y almacenamiento. A medida que aumenta el valor de n, el número de combinaciones posibles crece exponencialmente.

Actividad en clase

En equipos de, máximo 3 personas, haz la tabla de frecuencias y probabilidades para el siguiente texto:

> El aprendizaje automático está transformando la industria. El aprendizaje está transformando los modelos de lenguaje. Los modelos de lenguaje son un claro de revolución tecnológica. La industria de la revolución tecnológica está cambiando con el aprendizaje automático. El aprendizaje está transformando todo.

Section 3

Generalizaciones

Skip-grams

Un ***k*-skip-*n*-gram** es una subsecuencia de longitud n en la que los tokens aparecen a una distancia k como máximo entre sí.

the rain in Spain falls mainly on the plain

El conjunto de 1-skip-2-grams incluye todos los bigramas, además:

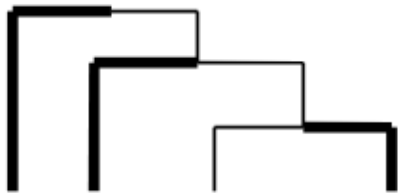
the in	rain Spain	in falls
Spain mainly	falls on	mainly the
on plain		

n-gramas sintácticos

A diferencia de los *n*-gramas donde las subsecuencias se toman en el orden en el que aparecen en el texto, en los ***n*-gramas sintácticos** los vecinos se toman siguiendo las relaciones sintácticas de los árboles de dependencia sintáctica.

eat with wooden spoon eat with metallic spoon

eat with wooden spoon



eat with metallic spoon

¿Cuántos bigramas y bigramas sintáticos en común tienen?

Section 4

Evaluando Modelos de Lenguaje

Evaluando Modelos de Lenguaje

Hay dos maneras de evaluar un modelo de Lenguaje:

- La **evaluación extrínseca** es la evaluación del desempeño del modelo en la tarea particular para la cual está siendo entrenado. La evaluación extrínseca es la única forma de saber si una mejora concreta de un componente va a ayudar realmente a la tarea que se está realizando.
- La **evaluación instrínseca** mide la calidad del modelo independientemente de la tarea o aplicación del modelo. Algunos ejemplos son: Entropía, Perplejidad, etc.

Perplejidad

La **perplejidad** es una métrica para evaluar el rendimiento de un modelo de lenguaje. Mide la incertidumbre del modelo para predecir la próxima palabra en una secuencia. Cuanto menor sea la perplejidad, mayor será la capacidad del modelo para predecir la palabra siguiente.

$$\begin{aligned} \mathrm{Pp}(W) &= \sqrt[N]{\frac{1}{P(w_1 w_2 \cdots w_N)}} \\ &= \sqrt[N]{\prod_i^N \frac{1}{P(w_i|w_{i-1})}} \end{aligned}$$

W es la secuencia entera de palabras de un conjunto de prueba.

Perplejidad: un ejemplo extremo

Considerar textos escritos en AAVE (African American Vernacular English):

> Bored af den my phone finna die

> Ah dont know what homey be doin.

¿Cómo sería la perplejidad de un modelo de lenguaje de *n*-gramas en estos textos de prueba?

Perplejidad

- Al calcular la perplejidad, el modelo de *n*-gramas debe construirse sin ningún conocimiento del conjunto de prueba. De otra forma la perplejidad puede ser artificialmente baja.
- La perplejidad de dos modelos lingüísticos sólo es comparable si utilizan vocabularios idénticos.
- Una mejora en la perplejidad (m. intrínseca) no garantiza una mejora del rendimiento de una tarea de PLN como el reconocimiento del habla o la clasificación (m. extrínseca). Sin embargo, la perplejidad suele estar correlacionada con dichas mejoras.

Métricas de evaluación de NLP

Métricas de evaluación de NLP

Métricas son para evaluar cualquier tipo de generación de texto, como traducción, resumenes, etc.

¿Qué miden fundamentalmente?

- Similitud entre texto generado automáticamente y texto de referencia
- Calidad de la generación comparada con un *gold standard*
- Qué tan bien nuestro modelo reproduce el comportamiento humano

Problema central: ¿Cómo medir objetivamente si un texto generado automáticamente es "bueno"?

Section 5

Aspectos adicionales

Suavizado: El problema de la dispersión

Si un *n*-grama aparece un número suficiente de veces, podemos tener una buena estimación de su probabilidad. Sin embargo, es probable que falten algunas secuencias de palabras perfectamente aceptables. Tendremos casos de *n-gramas de probabilidad 0* que deberían tener probabilidad distinta de 0.

denied the allegations:	5
denied the speculation:	2
denied the rumors:	1
denied the report:	1

denied the offer:	0
denied the loan:	0

Ejemplo

Suavizado

En los modelos de lenguaje estadísticos, como los n-gramas, a menudo nos encontramos con una secuencia de palabras que nunca apareció en el texto de entrenamiento. Si le asignamos una probabilidad de 0, el modelo se *rompe* porque cualquier oración que contenga esa secuencia tendrá una probabilidad total de 0. El suavizado es la técnica para evitar este problema, asignando una pequeña parte de la probabilidad a eventos no vistos.

Suavizado: Palabras desconocidas

¿Qué pasa con palabras que nunca ha visto en el entrenamiento?

En un sistema de **vocabulario cerrado** el conjunto de prueba no contiene palabras desconocidas. En un sistema de **vocabulario abierto**, tenemos que lidiar con palabras que no hemos visto antes, a las que llamaremos **palabras fuera de vocabulario (OOV)**. Podemos lidiar con estas palabras desconocidas eañadiendo una pseudopalabra llamada $\langle$UNK$\rangle$.

Palabras desconocidas

Hay dos estrategias:

- Convertir un sistema abierto en uno cerrado:
  - Escoger un vocabulario fijo.
  - Convertir cualquier palabra OOV en $\langle \text{UNK} \rangle$.
  - Estimar las probabilidades para $\langle \text{UNK} \rangle$ de la forma usual, como si fuera una palabra *normal*.
- Crear un vocabulario fijo implícito. Reemplazamos palabras por $\langle \text{UNK} \rangle$ en el entrenamiento basándonos en su frecuencia.

Suavizado

- Laplace Smoothing

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

- Add-k smoothing
- Kneser-Ney Smoothing
- ...

https://www.nltk.org/api/nltk.lm.smoothing.html

Section 6

Algunas aplicaciones

Google *n*-grams

- El Visor de *n*-gramas de Google es un motor de búsqueda que traza las frecuencias de *n*-gramas encontrados en fuentes impresas publicadas entre 1500 y 2022.
  Algunos ejemplos: 1, 2, 3
- Google Research pusó disponible un corpus grande de *n*-gramas. Incluye *n*-gramas que ocurren al menos 40 veces en una secuencia de 1,024,908,267,229 palabras.

Una aplicación: Análisis de Sentimientos

Una aplicación: Generación y predicción de texto