Modelos matemáticos del lenguaje natural (p.1 pre-2022)

Mauricio Velasco*
Centro de Matemáticas (CMAT)
Facultad de Ciencias
Universidad de la Republica

Julio 2025

Qué es un modelo de lenguaje?

El objetivo de un **modelo de lenguaje** *es capturar los patrones estadísticos del lenguaje natural.*

Qué es un modelo de lenguaje?

El objetivo de un **modelo de lenguaje** *es capturar los patrones estadísticos del lenguaje natural.*

Intuitivamente esto significa tener un mecanismo que nos permita decidir si una **sucesión de palabras** cualquiera es o no es una frase estadisticamente correcta y estimar su grado de plausibilidad.

Qué es un modelo de lenguaje?

El objetivo de un **modelo de lenguaje** *es capturar los patrones estadísticos del lenguaje natural.*

Intuitivamente esto significa tener un mecanismo que nos permita decidir si una **sucesión de palabras** cualquiera es o no es una frase estadisticamente correcta y estimar su grado de plausibilidad.

Concretamente, queremos construir una caja negra $\mathbb{P}$ que nos permita cuantificar la probabilidad de que frases como f ="el perro juega con la pelota" sean lenguaje y que nos diga que esta es mayor que la probabilidad de que la frase g="perro juega pelota perro ladra" sea lenguaje.

Qué es un modelo de lenguaje?

El objetivo de un **modelo de lenguaje** *es capturar los patrones estadísticos del lenguaje natural.*

Intuitivamente esto significa tener un mecanismo que nos permita decidir si una **sucesión de palabras** cualquiera es o no es una frase estadisticamente correcta y estimar su grado de plausibilidad.

Concretamente, queremos construir una caja negra $\mathbb{P}$ que nos permita cuantificar la probabilidad de que frases como $f =$"el perro juega con la pelota" sean lenguaje y que nos diga que esta es mayor que la probabilidad de que la frase $g=$"perro juega pelota perro ladra" sea lenguaje.

$$\mathbb{P}(f) >> \mathbb{P}(g)$$

P A R T E I:

Modelos probabilísticos del lenguaje.

Qué es un modelo de lenguaje?

Fijamos un **vocabulario** W que es un conjunto finito de símbolos (palabras) y una longitud $L \in \mathbb{N}$.

Definición. (Shannon, 1949)

Un **modelo de Lenguaje** *es una distribución de probabilidad en* W^L*, es decir una función* $\mathbb{P} : W^L \to \mathbb{R}_+$ *con* $\sum_{f \in W^L} \mathbb{P}(f) = 1$.

Ejemplo:

En un idioma cualquiera el vocabulario usual es del orden de 100.000 *palabras (*600*k Oxford dictionary) luego el dominio de* $\mathbb{P}$ *tiene tamaño* 100.000^L *y no podemos describir* $\mathbb{P}$ *como una tabla salvo en casos muy pequeños.*

frase	*probabilidad*
$w_1 \ldots w_L$	$\mathbb{P}(w_1 w_2 \ldots w_L)$

Cómo describir un modelo de lenguaje?

Recuerde que si $\mathcal{H} \subseteq W^L$ es un conjunto con $\mathbb{P}(\mathcal{H}) \neq 0$, entonces para cada frase $w_1 \dots w_L$ se define la **probabilidad condicional**

$$\mathbb{P}(w_1 \dots w_L | \mathcal{H}) = \frac{\mathbb{P}(w_1 \dots w_L \cap \mathcal{H})}{\mathbb{P}(\mathcal{H})}$$

Cómo describir un modelo de lenguaje?

Recuerde que si $\mathcal{H} \subseteq W^L$ es un conjunto con $\mathbb{P}(\mathcal{H}) \neq 0$, entonces para cada frase $w_1 \dots w_L$ se define la **probabilidad condicional**

$$\mathbb{P}(w_1 \dots w_L | \mathcal{H}) = \frac{\mathbb{P}(w_1 \dots w_L \cap \mathcal{H})}{\mathbb{P}(\mathcal{H})}$$

La función $\mathbb{P}(\bullet | \mathcal{H}) : W^L \to \mathbb{R}$ es otro modelo de lenguaje (i.e. otra distribucion de probabilidad) en frases de la misma longitud con el mismo vocabulario. Lo llamamos el **modelo condicional de lenguaje dado el contexto** $\mathcal{H}$.

Cómo describir un modelo de lenguaje?

Si $x = x_1 \cdot x_2 \cdot \cdots \cdot x_L$ es una secuencia y $\ell \leq L$ un entero, definimos $x_{\leq \ell} := x_1 \cdot \cdots \cdot x_\ell$.

Cómo describir un modelo de lenguaje?

Si $x = x_1 \cdot x_2 \cdot \cdots \cdot x_L$ es una secuencia y $\ell \leq L$ un entero, definimos $x_{\leq \ell} := x_1 \cdot \cdots \cdot x_\ell$.

Cómo describir $p(x)$?

Cómo describir un modelo de lenguaje?

Si $x = x_1 \cdot x_2 \cdot \cdots \cdot x_L$ es una secuencia y $\ell \leq L$ un entero, definimos $x_{\leq \ell} := x_1 \cdot \cdots \cdot x_\ell$.

Cómo describir $p(x)$?

Usando la **ley del producto** *podemos escribir*

$$p(x) = p(x_L | x_{\leq L-1}) p(x_{\leq L-1})$$

Cómo describir un modelo de lenguaje?

Si $x = x_1 \cdot x_2 \cdot \cdots \cdot x_L$ es una secuencia y $\ell \leq L$ un entero, definimos $x_{\leq \ell} := x_1 \cdot \cdots \cdot x_\ell$.

Cómo describir $p(x)$?

Usando la **ley del producto** *podemos escribir*

$$p(x) = p(x_L | x_{\leq L-1}) p(x_{\leq L-1})$$

y de manera iterada...

Cómo describir un modelo de lenguaje?

Si $x = x_1 \cdot x_2 \cdot \cdots \cdot x_L$ es una secuencia y $\ell \leq L$ un entero, definimos $x_{\leq \ell} := x_1 \cdot \cdots \cdot x_\ell$.

Cómo describir $p(x)$?

Usando la **ley del producto** *podemos escribir*

$$p(x) = p(x_L | x_{\leq L-1}) p(x_{\leq L-1})$$

y de manera iterada...

$$= p(x_L | x_{\leq L-1}) p(x_{m-1} | x_{\leq L-2}) p(x_{\leq L-2}) =$$

$$= p(x_L | x_{\leq L-1}) p(x_{L-1} | x_{\leq L-2}) \dots p(x_2 | x_{\leq 1}) p(x_{\leq 1})$$

La probabilidad en sucesiones de longitud L es un **producto de probabilidades condicionales** $p(x_m | x_{\leq m-1})$ sobre el vocabulario de interés dado el contexto.

Cómo describir un modelo de lenguaje?

$$p(x) = p(x_L | x_{\leq L-1}) p(x_{L-1} | x_{\leq L-2}) \dots p(x_2 | x_{\leq 1}) p(x_{\leq 1})$$

Cómo describir un modelo de lenguaje?

$$p(x) = p(x_L|x_{\leq L-1})p(x_{L-1}|x_{\leq L-2})\dots p(x_2|x_{\leq 1})p(x_{\leq 1})$$

Típicamente se asume que el **working memory** es limitado, es decir existe una **longitud de contexto** ℓ fija tal que para todo m

$$p(x_m|x_{\leq m-1}) = p(x_m|x_{[m-\ell, m-1]})$$

Cómo describir un modelo de lenguaje?

$$p(x) = p(x_L|x_{\leq L-1})p(x_{L-1}|x_{\leq L-2}) \dots p(x_2|x_{\leq 1})p(x_{\leq 1})$$

Típicamente se asume que el **working memory** es limitado, es decir existe una **longitud de contexto** ℓ fija tal que para todo m

$$p(x_m|x_{\leq m-1}) = p(x_m|x_{[m-\ell,m-1]})$$

La especificación de un modelo de lenguaje consiste en determinar las distribuciones condicionales

$$p\left(x_m|x_{[m-\ell,m-1]}\right)$$

del siguiente símbolo dado el contexto.

Usos de las distribuciones condicionales

Si tuviéramos las distribuciones condicionales

$$p\left(x_m | x_{[m-1-\ell, m-1]}\right)$$

para todo contexto, podriamos hacer dos tareas clave.

Usos de las distribuciones condicionales

Si tuviéramos las distribuciones condicionales

$$p\left(x_m | x_{[m-1-\ell, m-1]}\right)$$

para todo contexto, podriamos hacer dos tareas clave.

1. Podriamos usar el modelo para **generar** frases.

Usos de las distribuciones condicionales

Si tuviéramos las distribuciones condicionales

$$p\left(x_m | x_{[m-1-\ell, m-1]}\right)$$

para todo contexto, podriamos hacer dos tareas clave.

1. Podriamos usar el modelo para **generar** frases.
2. Podriamos, dado un conjunto $\mathcal{D}$ de frases, medir cuán probable es que nuestro modelo las genere.

Usos de las distribuciones condicionales

Si tuviéramos las distribuciones condicionales

$$p\left(x_m | x_{[m-1-\ell,m-1]}\right)$$

para todo contexto, podriamos hacer dos tareas clave.

1. Podriamos usar el modelo para **generar** frases.
2. Podriamos, dado un conjunto $\mathcal{D}$ de frases, medir cuán probable es que nuestro modelo las genere. Esta es la **verosimilitud del modelo dado** $\mathcal{D}$ y es la clave para entender el **entrenamiento** de redes neuronales.

Distribuciones condicionales para generar

Como continuar la frase "el perro rojo corrió "?

Distribuciones condicionales para generar

Como continuar la frase "el perro rojo corrió "?

1. Calculamos el vector $\mathbb{P}(w|\text{el perro rojo corrió})$

Distribuciones condicionales para generar

Como continuar la frase "el perro rojo corrió "?

1. Calculamos el vector $\mathbb{P}(w|\text{el perro rojo corrió})$
2. Ordenamos este vector de mayor a menor

palabra	probabilidad
rápidamente	0.7
hacia	0.2
⋮	⋮
cabizbajo	0.003
lentamente	0.001
⋮	⋮
perro	0.000001
⋮	⋮

Distribuciones condicionales para generar

Como continuar la frase "el perro rojo corrió "?

1. Calculamos el vector $\mathbb{P}(w|\text{el perro rojo corrió})$
2. Ordenamos este vector de mayor a menor

palabra	probabilidad
rápidamente	0.7
hacia	0.2
⋮	⋮
cabizbajo	0.003
lentamente	0.001
⋮	⋮
perro	0.000001
⋮	⋮

3. Seleccionamos una palabra de alta probabilidad

Distribuciones condicionales para generar

Como continuar la frase "el perro rojo corrió "?

1. Calculamos el vector $\mathbb{P}(w|\text{el perro rojo corrió})$
2. Ordenamos este vector de mayor a menor

palabra	probabilidad
rápidamente	0.7
hacia	0.2
⋮	⋮
cabizbajo	0.003
lentamente	0.001
⋮	⋮
perro	0.000001
⋮	⋮

3. Seleccionamos una palabra de alta probabilidad "el perro rojo corrió **hacia**"

Distribuciones condicionales para generación

Y luego repetimos, considerando como contexto una nueva ventana que contenga la palabra recien generada.

Distribuciones condicionales para generación

Y luego repetimos, considerando como contexto una nueva ventana que contenga la palabra recien generada. Para seguir extendiendo "el perro rojo corrió hacia" analizamos el vector de probabilidad

$$\mathbb{P}(w|\text{perro rojo corrió hacia})$$

y asi sucesivamente...

Distribuciones condicionales para generación

Y luego repetimos, considerando como contexto una nueva ventana que contenga la palabra recien generada. Para seguir extendiendo "el perro rojo corrió hacia" analizamos el vector de probabilidad

$$\mathbb{P}(w|\text{perro rojo corrió hacia})$$

y asi sucesivamente...

Pregunta.

Cómo empezar la generación?

Distribuciones condicionales para generación

Y luego repetimos, considerando como contexto una nueva ventana que contenga la palabra recien generada. Para seguir extendiendo "el perro rojo corrió hacia" analizamos el vector de probabilidad

$$\mathbb{P}(w|\text{perro rojo corrió hacia})$$

y asi sucesivamente...

Pregunta.

Cómo empezar la generación?

Distribuciones condicionales para generación

Y luego repetimos, considerando como contexto una nueva ventana que contenga la palabra recien generada. Para seguir extendiendo "el perro rojo corrió hacia" analizamos el vector de probabilidad

$$\mathbb{P}(w|\text{perro rojo corrió hacia})$$

y asi sucesivamente...

Pregunta.

Cómo empezar la generación?

- La habilidad de especificar una frase inicial es una de las grandes virtudes del proceso de generación mediante probabilidades condicionales.

Distribuciones condicionales para generación

Y luego repetimos, considerando como contexto una nueva ventana que contenga la palabra recien generada. Para seguir extendiendo "el perro rojo corrió hacia" analizamos el vector de probabilidad

$$\mathbb{P}(w|\text{perro rojo corrió hacia})$$

y asi sucesivamente...

Pregunta.

Cómo empezar la generación?

- La habilidad de especificar una frase inicial es una de las grandes virtudes del proceso de generación mediante probabilidades condicionales.
- Típicamente hay un token 0 para "padding" que nos permite pensar que una frase muy corta es simplemente la frase original con varios 0 al principio para que se vuelva de longitud L. Concretamente "el perro" = "0 0 el perro"

Verosimilitud

Dada una gran cantidad de frases $\mathcal{D} = d_1 d_2 \ldots d_N$.

Pregunta.

Qué tan probable es que nuestro modelo las hubiera generado?

Verosimilitud

Dada una gran cantidad de frases $\mathcal{D} = d_1 d_2 \ldots d_N$.

Pregunta.

Qué tan probable es que nuestro modelo las hubiera generado?

La primera palabra d_1 se habria generado con probabilidad $\mathbb{P}(d_1)$

Verosimilitud

Dada una gran cantidad de frases $\mathcal{D} = d_1 d_2 \ldots d_N$.

Pregunta.

Qué tan probable es que nuestro modelo las hubiera generado?

La primera palabra d_1 se habria generado con probabilidad $\mathbb{P}(d_1)$
Las primeras dos $d_1 d_2$ con probabilidad

$$\mathbb{P}(d_1)\mathbb{P}(d_2|d_1)$$

Verosimilitud

Dada una gran cantidad de frases $\mathcal{D} = d_1 d_2 \ldots d_N$.

Pregunta.

Qué tan probable es que nuestro modelo las hubiera generado?

La primera palabra d_1 se habria generado con probabilidad $\mathbb{P}(d_1)$
Las primeras dos $d_1 d_2$ con probabilidad

$$\mathbb{P}(d_1)\mathbb{P}(d_2|d_1)$$

Las primeras tres $d_1 d_2 d_3$ con probabilidad

$$\mathbb{P}(d_1)\mathbb{P}(d_2|d_1)\mathbb{P}(d_3|d_1 d_2)$$

Verosimilitud

Dada una gran cantidad de frases $\mathcal{D} = d_1 d_2 \ldots d_N$.

Pregunta.

Qué tan probable es que nuestro modelo las hubiera generado?

La primera palabra d_1 se habria generado con probabilidad $\mathbb{P}(d_1)$
Las primeras dos $d_1 d_2$ con probabilidad

$$\mathbb{P}(d_1)\mathbb{P}(d_2|d_1)$$

Las primeras tres $d_1 d_2 d_3$ con probabilidad

$$\mathbb{P}(d_1)\mathbb{P}(d_2|d_1)\mathbb{P}(d_3|d_1 d_2)$$

Los datos observados con probabilidad

$$V = \prod_{j=1}^{N} \mathbb{P}(d_j|d_{<j}) = \prod_{j=1}^{N} \mathbb{P}(d_j|d_{[j-1-\ell,j-1]})$$

este numero se llama la **verosimilitud del modelo** *(dado $\mathcal{D}$).*

Verosimilitud

La verosimilitud es un concepto extremadamente importante. **Si tenemos un conjunto de datos** $\mathcal{D}$ y varios modelos disponibles $\mathbb{P}_\theta$, $\theta \in \Theta$ podemos decir que θ_1 es mejor que θ_2 si

$$\prod_{j=1}^{N} \mathbb{P}_{\theta_1}(d_j | d_{[j-1-\ell, j-1]}) > \prod_{j=1}^{N} \mathbb{P}_{\theta_2}(d_j | d_{[j-1-\ell, j-1]})$$

Verosimilitud

La verosimilitud es un concepto extremadamente importante. **Si tenemos un conjunto de datos** $\mathcal{D}$ y varios modelos disponibles $\mathbb{P}_{\theta}$, $\theta \in \Theta$ podemos decir que θ_1 es mejor que θ_2 si

$$\prod_{j=1}^{N} \mathbb{P}_{\theta_1}(d_j | d_{[j-1-\ell, j-1]}) > \prod_{j=1}^{N} \mathbb{P}_{\theta_2}(d_j | d_{[j-1-\ell, j-1]})$$

$$V(\theta_1 | \mathcal{D}) > V(\theta_2 | \mathcal{D})$$

Verosimilitud

La verosimilitud es un concepto extremadamente importante. **Si tenemos un conjunto de datos** $\mathcal{D}$ y varios modelos disponibles $\mathbb{P}_\theta$, $\theta \in \Theta$ podemos decir que θ_1 es mejor que θ_2 si

$$\prod_{j=1}^{N} \mathbb{P}_{\theta_1}(d_j | d_{[j-1-\ell, j-1]}) > \prod_{j=1}^{N} \mathbb{P}_{\theta_2}(d_j | d_{[j-1-\ell, j-1]})$$

$$V(\theta_1 | \mathcal{D}) > V(\theta_2 | \mathcal{D})$$

Equivalentemente, si las probabilidades condicionales nunca valen cero, podemos decir que θ_1 es mejor que θ_2 en $\mathcal{D}$ si

$$\log(V(\theta_1 | \mathcal{D})) > \log(V(\theta_2 | \mathcal{D}))$$

Verosimilitud y Entrenamiento

Ahora construiremos una familia de modelos $\mathbb{P}_\theta$ dependiendo de un vector de parámetros $\theta \in \mathbb{R}^M$.
Asumiremos que tenemos un buen set de datos de entrenamiento $\mathcal{D}$ consistente de muchas frases (que para el algoritmo de entrenamiento son sólamente sucesiones de símbolos).

La maximizacion de log-verosimilitud nos da un mecanismo

$$\theta^* = \operatorname{argmax} (\log V(\theta|\mathcal{D}))$$

para seleccionar parámetros que permitan a nuestro modelo generar textos más parecidos a los datos.

Ese proceso de seleccion de parámetros en $\mathbb{R}^M$ se llama **entrenamiento** y, si el modelo tiene grandes cantidades de parametros puede ser muy demandante de recursos de computo.

Procedimiento de Cloze

Es muy fácil seleccionar algunas palabras y reemplazarlas por espacios en blanco. Pedimos al modelo que *llene los espacios en blanco* (Cloze procedure, una idea debida al psicólogo Wilson Taylor, 1953)

Procedimiento de Cloze

Es muy fácil seleccionar algunas palabras y reemplazarlas por espacios en blanco. Pedimos al modelo que *llene los espacios en blanco* (Cloze procedure, una idea debida al psicólogo Wilson Taylor, 1953)

Ejemplo:

- *(Original). El gato se sentó en el sofá.*

Procedimiento de Cloze

Es muy fácil seleccionar algunas palabras y reemplazarlas por espacios en blanco. Pedimos al modelo que *llene los espacios en blanco* (Cloze procedure, una idea debida al psicólogo Wilson Taylor, 1953)

Ejemplo:

- *(Original). El gato se sentó en el sofá.*
- *(Frase enmascarada). Completar: El gato se sentó en el YYY.*
- *(Respuesta deseada). Sofá*

Procedimiento de Cloze

Es muy fácil seleccionar algunas palabras y reemplazarlas por espacios en blanco. Pedimos al modelo que *llene los espacios en blanco* (Cloze procedure, una idea debida al psicólogo Wilson Taylor, 1953)

Ejemplo:

- *(Original). El gato se sentó en el sofá.*
- *(Frase enmascarada). Completar: El gato se sentó en el YYY.*
- *(Respuesta deseada). Sofá*
- *Este ejemplo contribuye a la función objetivo del entrenamiento aportando el término*

$$\log\left(p_\theta(\mathrm{Sofa}|\textit{El gato se sentó en el})\right)$$

Procedimiento de Cloze

Es muy fácil seleccionar algunas palabras y reemplazarlas por espacios en blanco. Pedimos al modelo que *llene los espacios en blanco* (Cloze procedure, una idea debida al psicólogo Wilson Taylor, 1953)

Ejemplo:

- *(Original). El gato se sentó en el sofá.*
- *(Frase enmascarada). Completar: El gato se sentó en el YYY.*
- *(Respuesta deseada). Sofá*
- *Este ejemplo contribuye a la función objetivo del entrenamiento aportando el término*

$$\log\left(p_\theta(\mathrm{Sofa} \mid \textit{El gato se sentó en el})\right)$$

Procedimiento de Cloze

Pedimos al modelo que aprenda a *llenar los espacios en blanco* (Cloze procedure, una idea debida al psicólogo Wilson Taylor, 1953)

Más concretamente, si $\mathcal{D}$ es nuestra colección de datos, queremos maximizar la probabilidad de observarlos, lo cual es equivalente al **maximum log-likelihood estimation**

$$\max_{\theta} \left(\sum_{(d_m | d_{[m-\ell, m-1]}) \in \mathcal{D}} \log \left(p_\theta(d_m | d_{[m-\ell, m-1]}) \right) \right)$$

Procedimiento de Cloze

Pedimos al modelo que aprenda a *llenar los espacios en blanco* (Cloze procedure, una idea debida al psicólogo Wilson Taylor, 1953)

Más concretamente, si $\mathcal{D}$ es nuestra colección de datos, queremos maximizar la probabilidad de observarlos, lo cual es equivalente al **maximum log-likelihood estimation**

$$\max_{\theta}\left(\sum_{(d_m|d_{[m-\ell,m-1]})\in\mathcal{D}} \log\left(p_\theta(d_m|d_{[m-\ell,m-1]})\right)\right)$$

En la práctica se usan modelos $p_\theta(d)$ que sean **funciones diferenciables** de θ y se busca un **minimo local** de la funcion objetivo mediante descenso del gradiente (estocástico SGD).

Un modelo de Lenguaje entiende el lenguaje?

Contrary to how it may seem when we observe its output, a LM is a system for haphazardly stitching together sequences of linguistic forms it has observed in its vast training data, according to probabilistic information about how they combine, but without any reference to meaning: a **stochastic parrot**. **Emily Bender (Directora del Laboratorio de Linguística computacional de la Univ. Washington)**

Un modelo de Lenguaje entiende el lenguaje?

> *Contrary to how it may seem when we observe its output, a LM is a system for haphazardly stitching together sequences of linguistic forms it has observed in its vast training data, according to probabilistic information about how they combine, but without any reference to meaning: a* **stochastic parrot***.* **Emily Bender (Directora del Laboratorio de Linguística computacional de la Univ. Washington)**

Un LLM no es otra cosa que un algoritmo de autocompletado grande. No entiende nada, sólo simula el entendimiento a través de lenguaje.

No obstante, los modelos probabilístico del lenguaje son una tecnología importante con algunas capacidades interesantes.

Usos de los modelos de lenguaje

No obstante, los modelos probabilístico del lenguaje son una tecnología importante con algunas capacidades interesantes.

1. Permiten distribuciones condicionales mas sofisticadas del tipo $\mathbb{P}(x_m | x_{[m-1-\ell,m-1]}, y_{[\ell]})$ donde la condición depende de otras variables.

Usos de los modelos de lenguaje

No obstante, los modelos probabilístico del lenguaje son una tecnología importante con algunas capacidades interesantes.

1. Permiten distribuciones condicionales mas sofisticadas del tipo $\mathbb{P}(x_m | x_{[m-1-\ell, m-1]}, y_{[\ell]})$ donde la condición depende de otras variables.Esto es esencial en aplicaciones como traducción.

Usos de los modelos de lenguaje

No obstante, los modelos probabilístico del lenguaje son una tecnología importante con algunas capacidades interesantes.

1. Permiten distribuciones condicionales mas sofisticadas del tipo $\mathbb{P}(x_m | x_{[m-1-\ell, m-1]}, y_{[\ell]})$ donde la condición depende de otras variables.Esto es esencial en aplicaciones como traducción.
2. El proceso de generación autoregresiva parece creativo, al menos hasta un punto, al mezclar patrones existentes de maneras novedosas.

Usos de los modelos de lenguaje

No obstante, los modelos probabilístico del lenguaje son una tecnología importante con algunas capacidades interesantes.

1. Permiten distribuciones condicionales mas sofisticadas del tipo $\mathbb{P}(x_m | x_{[m-1-\ell, m-1]}, y_{[\ell]})$ donde la condición depende de otras variables.Esto es esencial en aplicaciones como traducción.
2. El proceso de generación autoregresiva parece creativo, al menos hasta un punto, al mezclar patrones existentes de maneras novedosas.
3. A traves del aprendizaje de patrones estadísticos los LMs desarrollan una representación interna "geométrica" de los datos que nos permite hacer búsquedas y comparaciones semánticas.

P A R T E II:

Introducción a modelos neuronales de lenguaje.

Familias de modelos:

Queremos construir

$$p_\theta(x_m | x_{[m-\ell, m-1]})$$

Familias de modelos:

Queremos construir

$$p_\theta(x_m|x_{[m-\ell,m-1]})$$

familias de **distribuciones de probabilidad sobre** W que dependen del contexto $x_{[m-\ell,m-1]} := (x_{m-\ell}, \dots, x_{m-1})$.

Familias de modelos:

Queremos construir

$$p_\theta(x_m | x_{[m-\ell, m-1]})$$

familias de **distribuciones de probabilidad sobre** W que dependen del contexto $x_{[m-\ell,m-1]} := (x_{m-\ell}, \ldots, x_{m-1})$.

Una función útil para construir distribuciones de probabilidad es la **función de Bolzmann** *(ó softmax)* $B : \mathbb{R}^N \to \Delta(\mathbb{R}^N)$ *dada por la fórmula*

$$\mathbb{B}(u_1, \ldots, u_N)_j := \frac{\exp(u_j)}{\sum_{i=1}^N (\exp(u_i))}$$

Familias de modelos:

Queremos construir

$$p_\theta(x_m | x_{[m-\ell, m-1]})$$

familias de **distribuciones de probabilidad sobre** W que dependen del contexto $x_{[m-\ell,m-1]} := (x_{m-\ell}, \dots, x_{m-1})$.

Una función útil para construir distribuciones de probabilidad es la **función de Bolzmann** *(ó softmax)* $B : \mathbb{R}^N \to \Delta(\mathbb{R}^N)$ *dada por la fórmula*

$$\mathbb{B}(u_1, \dots, u_N)_j := \frac{\exp(u_j)}{\sum_{i=1}^N (\exp(u_i))}$$

Asumiremos:

$$p(x_m = w_j | x_{[m-\ell,m-1]}) = \mathbb{B}\left(F_\theta(x_{[m-\ell,m-1]})\right)_j$$

con $F_\theta : W^\ell \to \mathbb{R}^{|W|}$ *una función que depende de parámetros* θ.

El modelo

Asumiremos:

$$p(x_m = w_j | x_{[m-\ell,m-1]}) = \mathbb{B}\left(F_\theta(x_{[m-\ell,m-1]})\right)_j$$

con $F_\theta : W^\ell \to \mathbb{R}^{|W|}$ *una función que depende de parámetros* θ.

Para construir nuestro modelo paramétrico podríamos utilizar cualquier familia de funciones $F_\theta : W^\ell \to \mathbb{R}^{|W|}$.

El modelo

Asumiremos:

$$p(x_m = w_j | x_{[m-\ell,m-1]}) = \mathbb{B}\left(F_\theta(x_{[m-\ell,m-1]})\right)_j$$

con $F_\theta : W^\ell \to \mathbb{R}^{|W|}$ *una función que depende de parámetros* θ.

Para construir nuestro modelo paramétrico podríamos utilizar cualquier familia de funciones $F_\theta : W^\ell \to \mathbb{R}^{|W|}$.

Lo importante *es que queremos seleccionar el vector de parámetros* θ *de tal forma que nuestras predicciones se parezcan a las sucesiones de símbolos observados.*

El modelo

Asumiremos:

$$p(x_m = w_j | x_{[m-\ell,m-1]}) = \mathbb{B}\left(F_\theta(x_{[m-\ell,m-1]})\right)_j$$

con $F_\theta : W^\ell \to \mathbb{R}^{|W|}$ una función que depende de parámetros θ.

Para construir nuestro modelo paramétrico podríamos utilizar cualquier familia de funciones $F_\theta : W^\ell \to \mathbb{R}^{|W|}$.

Lo importante *es que queremos seleccionar el vector de parámetros θ de tal forma que nuestras predicciones se parezcan a las sucesiones de símbolos observados.(***Entrenamiento***).*

Algunos modelos paramétricos

Asumiremos:

$$p(x_m = w_j | x_{[m-\ell, m-1]}) = \mathbb{B}\left(F_\theta(x_{[m-\ell, m-1]})\right)_j$$

con $F_\theta : W^\ell \to \mathbb{R}^{|W|}$ *una función que depende de parámetros* θ.

Word2Vec

Para empezar tomemos un contexto muy corto, solo la palabra anterior

$$p(x_m = w_j | x_{m-1} = w_s) = \mathbb{B}\left(F_\theta(w_s)\right)_j$$

Para empezar tomemos un contexto muy corto, solo la palabra anterior

$$p(x_m = w_j | x_{m-1} = w_s) = \mathbb{B}\left(F_\theta(w_s)\right)_j$$

y hagamos que F_θ sea la función más sencilla posible, mediante una **factorización de bajo rango** k.

Word2Vec

Para empezar tomemos un contexto muy corto, solo la palabra anterior

$$p(x_m = w_j | x_{m-1} = w_s) = \mathbb{B}\left(F_\theta(w_s)\right)_j$$

y hagamos que F_θ sea la función más sencilla posible, mediante una **factorización de bajo rango** k.
[2013 Mikolov, "Efficient Estimation of word representations in Vector Space"]

Word2Vec

Definimos

$$p(x_m = w_j | x_{m-1} = w_s) := \mathbb{B}\left(F_\theta(w_s)\right)_j$$

con $F_\theta(w_s) = U^T X(e_s)$

$$\mathbb{R}^{|W|} \xrightarrow{X} \mathbb{R}^k \xrightarrow{U^T} \mathbb{R}^{|W|}$$

y $\theta = (X, U) \in \mathbb{R}^{k \times |W|} \times \mathbb{R}^{k \times |W|}$.

Word2Vec

Definimos

$$p(x_m = w_j | x_{m-1} = w_s) := \mathbb{B}\left(F_\theta(w_s)\right)_j$$

con $F_\theta(w_s) = U^T X(e_s)$

$$\mathbb{R}^{|W|} \xrightarrow{X} \mathbb{R}^k \xrightarrow{U^T} \mathbb{R}^{|W|}$$

y $\theta = (X, U) \in \mathbb{R}^{k\times|W|} \times \mathbb{R}^{k\times|W|}$.

El modelo se entrena

$$\max \sum_{(w_j|w_s)\in\mathcal{D}} \log\left(\mathbb{B}\left(F_\theta(w_s)\right)_j\right)$$

mediante descenso por gradiente encontrando $\theta^* = (X^*, U^*)$.

Y. Goldberg and O.Levy word2Vec explained: Deriving Mikolov et al's Negative sampling word embedding method.

Word2Vec y la geometria del lenguaje

Una vez el entrenamiento termina extraemos la matriz $X^* \in \mathbb{R}^{k \times |W|}$

$$X^* = \begin{pmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_{|W|} \\ | & | & \cdots & | \end{pmatrix}$$

Word2Vec y la geometria del lenguaje

Una vez el entrenamiento termina extraemos la matriz $X^* \in \mathbb{R}^{k\times|W|}$

$$X^* = \begin{pmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_{|W|} \\ | & | & \cdots & | \end{pmatrix}$$

la j-ésima columna de X^* es una **representación geométrica** *del símbolo abstracto (palabra) e_j.*

Word2Vec y la geometria del lenguaje

Una vez el entrenamiento termina extraemos la matriz $X^* \in \mathbb{R}^{k\times|W|}$

$$X^* = \begin{pmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_{|W|} \\ | & | & \cdots & | \end{pmatrix}$$

la j-ésima columna de X^* es una **representación geométrica** *del símbolo abstracto (palabra) e_j.*

El entrenamiento hace que palabras con significados semejantes esten cerca en esa representación produciendo una **geometría-semántica** *de manera completamente automática.*

Word2Vec y la geometria del lenguaje

Word2Vec

http://epsilon-it.utu.fi/wv_demo/

Negative sampling en Word2Vec

Un modelo alternativo es pensar en el aprendizaje del lenguaje como un problema de clasificacion.

Negative sampling en Word2Vec

Un modelo alternativo es pensar en el aprendizaje del lenguaje como un problema de clasificacion.

Queremos aprender la distribucion condicional de la funcion indicadora $Z \in \{0,1\}$ *de "ser lenguaje",*

$$\mathbb{P}_\theta(Z = 1 | (w, c))$$

Negative sampling en Word2Vec

Un modelo alternativo es pensar en el aprendizaje del lenguaje como un problema de clasificacion.

Queremos aprender la distribucion condicional de la funcion indicadora $Z \in \{0, 1\}$ de "ser lenguaje",

$$\mathbb{P}_\theta(Z = 1|(w, c))$$

$$\mathbb{P}_\theta(Z = 0|(w, c)) = 1 - \mathbb{P}_\theta(Z = 1|(w, c))$$

Negative sampling en Word2Vec

Un modelo alternativo es pensar en el aprendizaje del lenguaje como un problema de clasificacion.

Queremos aprender la distribucion condicional de la funcion indicadora $Z \in \{0,1\}$ de "ser lenguaje",

$$\mathbb{P}_\theta(Z = 1|(w, c))$$

$$\mathbb{P}_\theta(Z = 0|(w, c)) = 1 - \mathbb{P}_\theta(Z = 1|(w, c))$$

Postulamos un modelo parametrico

$$\mathbb{P}_\theta(Z = 1|(w, c)) = \frac{1}{1 + e^{-v_c \cdot v_w}}$$

$$\mathbb{P}_\theta(Z = 0|(w, c)) = \frac{e^{-v_c \cdot v_w}}{1 + e^{-v_c \cdot v_w}} = \frac{1}{1 + e^{v_c \cdot v_w}}$$

Negative sampling en Word2Vec

Entrenar este modelo, es decir encontrar los valores de nuestras matrices de parámetros $\theta = \{v_w : w \in W\} \in \mathbb{R}^{k \times |W|}$ requiere:

1. Tener un conjunto grande de entrenamiento $(w, c) \in P$ de **instancias positivas**, que extraemos de texto y
2. Tener un conjunto grande de entrenamiento $(w, c) \in N$ de **instancias negativas**, que consiste de parejas $(w, c) \notin D$ que generamos aleatoriamente.
3. Dado un modelo podemos calcular su **verosimilitud** en nuestros datos.

$$V_\theta(\theta) = \prod_{(w,c)\in P} \mathbb{P}_\theta(Z = 1|(w, c)) \prod_{(w,c)\in N} \mathbb{P}_\theta(Z = 0|(w, c))$$

cuya maximizacion nos da un buen mecanismo de entrenamiento.

Negative sampling en Word2Vec

Dado un modelo podemos calcular su **verosimilitud** *en nuestros datos.*

$$V_\theta(\theta) = \prod_{(w,c)\in P} \mathbb{P}_\theta(Z = 1|(w,c)) \prod_{(w,c)\in N} \mathbb{P}_\theta(Z = 0|(w,c))$$

cuya maximizacion nos da un buen mecanismo de entrenamiento.

Negative sampling en Word2Vec

Dado un modelo podemos calcular su **verosimilitud** *en nuestros datos.*

$$V_\theta(\theta) = \prod_{(w,c)\in P} \mathbb{P}_\theta(Z=1|(w,c)) \prod_{(w,c)\in N} \mathbb{P}_\theta(Z=0|(w,c))$$

cuya maximizacion nos da un buen mecanismo de entrenamiento.

Tomando logaritmos esto es equivalente a maximizar sobre $\theta = \{v_w : w \in W\} \in \mathbb{R}^{k\times|W|}$

$$\sum_{(w,c)\in P} \log\left(\frac{1}{1+e^{-v_c\cdot v_w}}\right) + \sum_{(w,c)\in N} \log\left(\frac{1}{1+e^{v_c\cdot v_w}}\right)$$

Negative sampling en Word2Vec

Dado un modelo podemos calcular su **verosimilitud** *en nuestros datos.*

$$V_\theta(\theta) = \prod_{(w,c)\in P} \mathbb{P}_\theta(Z = 1|(w,c)) \prod_{(w,c)\in N} \mathbb{P}_\theta(Z = 0|(w,c))$$

cuya maximizacion nos da un buen mecanismo de entrenamiento.

Tomando logaritmos esto es equivalente a maximizar sobre $\theta = \{v_w : w \in W\} \in \mathbb{R}^{k\times|W|}$

$$\sum_{(w,c)\in P} \log\left(\frac{1}{1+e^{-v_c\cdot v_w}}\right) + \sum_{(w,c)\in N} \log\left(\frac{1}{1+e^{v_c\cdot v_w}}\right)$$

Ese proceso es conocido como **negative training**.

Usos de la funcion indicadora

Un modelo entrenado de la funcion indicadora Z se puede usar como **detector de anomalias**. Concretamente,

Una pareja $d_1 d_2$ puede considerarse **anomala** *(segun nuestro modelo) si el numero $\mathbb{P}_{\theta^*}(Z = 1|(d_2, d_1))$ es atipicamente bajo.*

Usos de la funcion indicadora

Un modelo entrenado de la funcion indicadora Z se puede usar como **detector de anomalias**. Concretamente,

Una pareja $d_1 d_2$ puede considerarse **anomala** *(segun nuestro modelo) si el numero $\mathbb{P}_{\theta^*}(Z = 1|(d_2, d_1))$ es atipicamente bajo.*

Esto es muy util, por ejemplo como mecanismo para detectar errores ortográficos o de redaccion.

Un caso concreto (JAMPI)

Desde 1993 el sistema de salud colombiano es un sistema publico-privado estructurado en tres niveles

- IPSs (clinicas, hospitales)
- EPSs (Sanitas, Compensar, etc.)
- ADREs (Sistema publico)

Este sistema permitio pasar de un cubrimiento del 17% de la poblacion (en 1990) a un 97% (en 2022) y aparece consistentemente ranqueado entre los mejores de america (en rankings WHO). Colombia tiene una poblacion de 40 millones de personas.

Un caso concreto (JAMPI)

- IPSs (clinicas, hospitales)
- EPSs (Sanitas, Compensar, etc.)
- ADRES (Sistema publico)

No obstante el sistema tiene cierta complejidad en sus procesos:

1. Los pacientes, afiliados a las EPSs reciben servicios de salud de las IPSs. Las EPSs pagan por estos servicios.
2. Las EPSs recobran a la ADRES los costos de salud que pagan a las IPSs o en algunos casos las IPSs cobran a la ADRES directamente.

El paso (2), llamado **recobros** *necesita que las IPSs y EPSs cobren a la ADRES sus cuentas médicas. Este proceso tiene gran probabilidad de error y se audita con mucho cuidado. Como mejorar (tanto la generacion como la auditoria)?*

Un caso concreto (JAMPI)

Pregunta.

Como mejorar la generación y auditoría de facturas?

- Insumos: Base de datos de **facturas** de una IPS. Cada factura contiene uno o varios codigos CUPS (Clasificación Única de Procedimientos en Salud) , los medicamentos (CUMS) e insumos correspondientes, incluyendo sus precios y cantidades.
- Objetivo es **disminución de glosas**: Dada la pre-factura de un procedimiento médico buscar de manera automática anomalías en el conjunto de procedimientos, medicamentos o insumos y/o en los precios / cantidades con la intención de disminuir las glosas.

Un caso concreto (JAMPI)

Dividimos el problema de aprendizaje en dos fases: el aprendizaje de los medicamentos e insumos que corresponden a cada procedimiento en salud y luego el aprendizaje sus precios.

Un caso concreto (JAMPI)

Dividimos el problema de aprendizaje en dos fases: el aprendizaje de los medicamentos e insumos que corresponden a cada procedimiento en salud y luego el aprendizaje sus precios.

La primera parte fase se resolvio mediante un modelo de lenguaje. Construimos, a partir de la base de datos de facturas un corpus de parejas (w, c) *de la forma* (CUM_1, CUP_1), (CUM_2, CUP_1), $(INSUMO_1, CUP_1)$ *con los insumos que aparecen en cada procedimiento medico y entrenamos un modelo tipo word2vec con negative sampling mediante maximizacion la verosimilitud en el dataset*

$$\mathbb{P}_{\theta^*}(Z = 1 | (w, c))$$

Un caso concreto (JAMPI)

Dividimos el problema de aprendizaje en dos fases: el aprendizaje de los medicamentos e insumos que corresponden a cada procedimiento en salud y luego el aprendizaje sus precios.

La primera parte fase se resolvio mediante un modelo de lenguaje. Construimos, a partir de la base de datos de facturas un corpus de parejas (w, c) *de la forma* (CUM_1, CUP_1), (CUM_2, CUP_1), $(INSUMO_1, CUP_1)$ *con los insumos que aparecen en cada procedimiento medico y entrenamos un modelo tipo word2vec con negative sampling mediante maximizacion la verosimilitud en el dataset*

$$\mathbb{P}_{\theta^*}(Z = 1|(w, c))$$

Se implementó un software (JAMPI) que chequea cada factura buscando items anomalos mediante las $\mathbb{P}_{\theta^*}$ (en uso desde 2022).