Lingüística computacional

Modelo del lenguaje

Objetivo: asignar probabilidades a una secuencia de palabras o dar una estimación probabilística de la siguiente palabra dada una secuencia.

Modelo de n-gramas

N-grama es una secuencia de n-palabras.

Problema → necesidad de n número elevado de muestras para el aprendizaje

➔ Se obtiene un número grande de parámetros:

- UNIGRAMA (Vocab=V == único estado)→ P(W)
- BIGRAMA (V^2 == Tantos estados como palabras en el vocab y las trnasiciones represemtam los bigramas) →P(W|W-1)
- TRIGRAMA V^3 (estados son secuencias de 2 palabras y las emisiones son los tri → P(W| W-1 W-2)

Estimación de MV

P(W|W-1) = P(W W-1)/ P(W)

Este criterio asigna una probabilidad 0 a los criterios o vistos → problemas de cobertura

Métodos de suavizado

Reservar una cierta cantidad de probabilidad descontada de los eventos que ya se han visto en el train para repartirla entre los eventos no observados.

Porqué usar métodos de suavizado

Debido al problema de sparse data.

Estos datos son los que nunca se han visto en el conjunto de entrenamiento, por lo cual su estimación máxima de verosimilitud de dicha probabilidad es 0. Por eso, si modificamos las probabilidades de máxima verosimilitud de las palabras que no han aparecido en el entrenamiento usando técnicas de smoothing ayudan a favorecen a las distribuciones que son 0 pero a su vez penalizan a las distribuciones grandes, aun así sirven para que una cadena de texto con una palabra que no ha aparecido nunca pueda tener una probabilidad distinta de 0.

Suavizado de Laplace

Es un método sencillo de suavizado que consiste en añadir 1 a todos los contadores de la matriz.

P*=c(wi wi-1) +1 / c(wi-1) + V , donde V es el vocabulario

Como estamos repartidiendo masa de probabilidad los contadores se reajustan:
C* = ((wi wi-1)+1) * w(i-1) / w(i-1) * V

Y para calcular que tanto difieren los contadores nuevos de los antiguos sin suavizar podemos calcular su ratio:
d = c*/ c

Interpolación lineal

El valor del suavizado se obtiene a apartir de la combinación lineal de los valores dados por una combinación lineal en función del trigrama o bigrama. Donde cada peso Lambda esta condicionado por el contexto del término anterior.

Estimador Good Turing

Este método de descuento se centra en que todo lo que se ha visto antes sirve para calcular aquello que no hemos visto aún.

C* = (c+1) * Nc / N , donde C es la clase c (0,1,2,3…) y Nc son el # de elemen tos vistos en la clase c

P* = Nc / N

Método Back Off

En resumen, este método lo que hace es que si ya se ha visto dicho n-grama se calcula su probabilidad, sino se ha visto lo que hace este método es retroceder a un n-grama de menor orden y si se ha visto este n-grama se calcula la probabilidad estimada, sino se vuelve a retroceder. Por lo que hace una búsqueda recursiva.

Diferencia entre métodos de descuento (LAPLACE; GT) y Back off / IL

La gran diferencia entre estos los métodos de suavizado es que los de descuento lo que hacen es reducir la probabilidad estimada de los n-gramas para evitar que sea 0, para prevenir problemas de sparse data. **(reservar masa de probabilidad para los que no hemos visto)**

En cuanto a los otros dos métodos, el Back off explica como distribuir la probabilidad de los n gramas no vistos retrocediendo a n-gramas de menor orden, cogiendo su probabilidad. Del mismo modo Int. Lineal lo que hace es que combina las distintas probabilidades y a estas les añade un peso en función de su frecuencia y función en el corpus. (**como distribuir la masa de probabilidad de los que no hemos visto)**

Método absolute discounting

Estos estimadores sacan termino fijo de descuento (D*) entonces este descuento va en función del count de las palabras, para count mayores el descuento es menor, ya que son las palabras más fiables y a la inversa con las palabras con menor count porque ya de todos modos no nos podemos fiar de ellas. Por eso, no afectara mucho el descuento a las palabras que nos podemos fiar, es una manera de ponderar el descuento para intentar tener los resultados más precisos.

Kneaser Ney Discounting

Este método se basa en que si se ha visto ya ese término en el vocabulario se calcula su probabilidad restándole el descuento. Y sino se ha visto se calcula la probabilidad de continuación de la palabra en función a los contextos más probables de aparecer en nuevos contextos. Porque puede haber muchas palabras con mayor frecuencia pero que no suelen ser probables en otros contextos.

Descuento Witten Bell

Se basa en que la palabra que no ha aparecido va a parecer en algún momento. Entonces se modela la probabilidad de los n-gramas no vistos con la probabilidad de aparecer.

Modelos basados en categorías

Se agrupan las palabras en categorías para reducir el número de parámetros a estimar y el número de muestras de aprendizaje.

Modelos dinámicos

Son los modelos para los que su estimación va cambiando a medida que se analiza el corpus del test.

Caché

Las palabras que aparecen tienden a volver a aparecer, es por eso que el modelo trabaja como una memoria a corto plazo y usa las frecuencias recientes para actualizar el modelo

Triggers

Trata de captar las correlaciones entre las secuencias de palabras del corpus de datos del entrenamiento.

Modelos gramaticales

Rev alg inside

Skip grams

Técnica basada en n-gramas que permite que algunas palabras sean saltadas

Factored LM

Las palabras son vectores con k factores, por lo que dentro de los vectores tenemos varia información como clases morfológicas, características de la lengua…

Modelos de lenguaje basados en n-gramas

Objetivo es calcular la probabilidad de una secuencia de tokens, calculándose a partir de un contexto (anterior palabra).

Inconvenientes:

- Se necesitan muchos datos para encontrar ocurrencias largas
- Por lo que conlleva a la asunción de Márkov
- Porcentaje muy elevado de n-gramas no vistos

Modelos de lenguaje Neuronales

Resuelven los problemas de los n-gramas:

- No requieren suavizado (gracias a las subwords)
- No hace falta recurrir a la asunción de Markow para trabajar con secuencias largas
- Puede generalizar en contextos de palabras parecidas (por ejemplo: tengo que darle comida al ___, en n-gramas solo se ha visto gato entonces asignaría esa palabra pero con MLN asignaría con una probabilidad del 50 % la palabra perro porque son 2 palabras que se parecen mucho en este contexto)

Hay distintos enfoques, modelos y arquitecturas.

PIPELINE:

1. Se alimenta la red con tokens del contexto (input)
2. Se genera la representación vectorial (embedding)
3. A partir del embedding se calcula la distribución de probabilidad de los tokens.

Embeddings incontextuales → tienen toda la información de un mismo token en un solo vector (banco: peces, dinero o sentarse), esto siempre aparece independientemente del contexto

Embeddings contextuales → seleccionan la información del contexto específico en el que aparece dicho token

Modelos de lenguaje neuronales causales

- Asignan una probabilidad a la secuencia de tokens pero la distribución de la probabilidad del siguiente token se calcula con una red neuronal
- Procesan el texto **unidireccionalmente** y así se calcula la probabilidad del siguiente token
- No es necesario considerar la asunción de Markow

NEURAL NETWORK LANGUAGE MODEL (diapo 10)

Se basa en el perceptrón multicapa.

Mantienen la asunción de Markow.

Es muy simple porque a partir de una ventana deslizante se van concatenando los embeddings incontextuales de los tokens de entrada para maximizar la probabilidad. Una vez entrenado puede usarse para calcular la probabilidad de un token dado los tokens previos, para calcula la secuencia de un texto o para generar texto.

RECURRENT NEURAL NETWORK LANGUAGE MODEL (diapo 13)

La historia se representa mediante conexiones recurrentes entre las neuronas (No asunción de Markow).

La diferencia entre el modelo anterior es que el embedding contextual de un token es la salida de la neurona de la RNN de ese token.

Tienen los mismos usos.

MODELOS CAUSALES MODERNOS

Cambian las RNN por Transoformers (decoder) y en lugar de palabras usan subwords para reducir el tamaño del conjunto de train trabajando solo con caracteres: token + izacion en lugar de tokenizacion.

Se usan las subwords porque trabajar con palabras es inviable ya que no se puede cubrir todo un vocabulario y si se pudiera se requerirían matrices de embeddings muy grandes. Esto reduce en gran medida las OOV.

GPT es una familia de modelos causales. GPT – 3 fue el primer modelo en lograr alcanzar el comportamiento de, sin embargo, no se debe de olvidar alinear los modelos con el feedback humano.

- Few shot: realizar tareas con datos de ejemplo
- Zero shot: realizar tareas sin datos de ejemplo

ChatGPT es la interfaz gráfica para hablar con un modelo GPT-3.5

Se observa que si escalamos el número de parámetros y tokens en proporciones muy similares obtenemos modelos óptimos.

DECODING

Se hace mediante sampling ya que para generar el sufijo más probable nos basamos en que los humanos no generan texto con máxima probabilidad sino buscan sorpresa, pero con coherencia (eliminar palabras redundantes, pronombres, repeticiones…)

Para ello usamos softmax con temperatura, en función de la temperatura que asignamos tenemos varios resultados

- Todo a uno (se elige solo una solución)
- Distribución por probabilidad
- Uniforme

Otra estrategia d decoding seria:

- Seleccionar las top k prefijos (de manera random)
- Top – p (en función de las probabilidades)

Evaluación

Hoy en día como los modelos son tan buenos debemos de fijarnos también en otras cosas a parte del rendimiento de modelo como es la toxicidad, el sesgo, la justicia…

COMPARATIVA MODELOS DE LENGUAJE CAUSALES Y N-GRAMAS

Ejemplo Cat/ Dog: Porque si solo se ve la palabra cat en el n-grama este solo elige cat porque es la que se ha visto durante el entrenamiento y como dog después del suavizado tiene una probabilidad muy baja no se elegiría al final, pero en MLC se podría elegir tb dog porque tienen contextos parecidos.

- Por eso, estos modelos permiten la **generalización entre contextos similares** la cual cosa no es posible con N-gramas
- Así como también se consiguen **generalizaciones en contextos no vistos**
- Pueden **resolver tareas one-shot y zero-shot**
- **No requieren técnicas de suavizado**
- **Pero son más costos y más lentos**

Modelos de lenguaje neuronales CONDICONALES

Los modelos causales estiman la probabilidad de una secuencia Y (P(Y)), en cambio las condiciones estiman la probabilidad de una target Y, P(Y|w)→ traducción y resumen automático y son encoder-decoder.

Denoising

Se puede entrenar un modelo condicional con objetivos de modelado de lenguaje para reconstruir el texto ruidoso y luego se puede fine tunear para otros usos.

Representación de palabras

Representación vectorial de las palabras donde las palabras parecidas están cerca en el espacio vectorial

Dispersa (Matriz de palabras)

Si sus vectores de contexto son similares las palabras son similares

Densa (Embedding)

Pueden ser contextuales(solo el significado basado en el contexto) e incontextuales (todos los significados en el vector)

Skip-gram y COW (diapo 52) → calcular la palabra central a través de las otras y calcular a través de la centras las otras palabras de la ventana de tamaño k

Palabras similares se representan en vectores similares y presentan relaciones aritméticas similares.

Bidireccional Enconder (BERT)

Transformer enconder entrenado para masked language, la idea esque aprende la lengua enmascarando muestras aleatorias y modela de manera direccional, con la información de la izq y la de la derecha. Después puede finetunearse para una tarea concreta, añadiendo capas adicionales sobre la última capa y reentrenando el corpus. (pre-training + fine-tuning = transfer learning) → entrenar para modelar el lenguaje y dps reentrenar para la tarea target.

RoBERta es igual, pero con un masking dinámico, por lo que cada vez que enmascaro uso una máscara distinta.

POS Tagging → Es el proceso de asignar a cada una de las palabras del texto su categoría gramatical, lo cual ayuda a desambiguar algunas de las palabras del texto.

Detección de entidades → las entidades son segmentos de la oración que tienen nombres de persona, organizaciones, lugares, fechas… (NER)

Word sense desambiguation → detección del sentido correcto de la frae