

Procesamiento de Lenguaje Natural

Apunte de Clases (Borrador)

Felipe Bravo Márquez

Felipe Bravo Márquez

Ilustración Portada por Paulette Filla

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN, UNIVERSIDAD DE CHILE

GITHUB.COM/DCCUCHILE/CC6205

Apuntes de clases del curso de Procesamiento de Lenguaje Natural de la Universidad de Chile.

El formato del apunte fue tomado del template de Jasmine Hao.

Borrador, 31 de octubre de 2023



Índice general

Prefacio	5	
1	Introducción	7
2	Modelo de Espacio Vectorial	9
3	Modelos de Lenguaje Probabilísticos	11
3.1	El Problema del Modelado del Lenguaje	11
3.1.1	¿Por qué queríramos hacer esto?	12
3.2	Regla de la Cadena en Modelos de Lenguaje	12
3.3	Mirada Predictiva	13
3.4	Mirada Generativa	14
3.5	Un Método Ingenuo	14
3.6	Procesos de Markov	15
3.6.1	Modelado de secuencias de longitud variable	15
3.7	Modelos de lenguaje de Trigramas	16
3.8	Evaluación de un modelo de lenguaje: Perplejidad	17
3.8.1	Intuición sobre la perplejidad	17
3.8.2	El trade-off entre sesgo y varianza	18
3.9	Interpolación Lineal	19
3.9.1	Estimación de los Valores λ	19

3.10	Métodos de Descuento	19
3.10.1	Modelos de Katz Back-Off (Bigramas)	20
3.11	Historia	22
3.12	Conclusiones	22
4	Clasificación de Texto y Naïve Bayes	25
5	Modelos Lineales	27
6	Redes Neuronales	29
7	Vectores de Palabra	31
8	Etiquetado de Secuencias	33
9	Redes Neuronales Convolucionales	35
10	Redes Neuronales Recurrentes	37
11	Modelos Secuencia a Secuencia	39
12	Arquitectura de Transformer	41
13	Grandes Modelos de Lenguaje	43



Prefacio

Este apunte es el resultado de cinco años de experiencia en la enseñanza del curso de Procesamiento de Lenguaje Natural en la Universidad de Chile¹. El objetivo es proporcionar una introducción a esta disciplina, centrándose en las técnicas y conceptos fundamentales. Se ha realizado un esfuerzo por lograr un equilibrio entre las técnicas tradicionales, como los modelos de lenguaje de N-gramas, Naive Bayes y los modelos ocultos de Markov (HMM), y los enfoques modernos basados en redes neuronales profundas, como los vectores de palabras, las redes neuronales recurrentes (RNN), los transformers y los grandes modelos de lenguaje. Sin embargo, es importante mencionar que existen temas relevantes que no se incluyen en este material, como el etiquetado de datos, las gramáticas, las técnicas de parsing, así como discusiones sobre tareas más específicas como el question answering.

El contenido se ha recopilado de diversas fuentes. Los temas relacionados con las redes neuronales se basan principalmente en el libro "Neural Network Methods for Natural Language Processing" de Goldberg [Goldberg, 2017]. Los temas no relacionados con las redes neuronales, como los modelos probabilísticos del lenguaje, Naive Bayes y HMM, se han tomado del curso de Michael Collins de Columbia [Collins, 2013] y del borrador de la tercera edición del libro de Dan Jurafsky y James H. Martin [Jurafsky and Martin, 2023]. Además, algunos capítulos se han adaptado de tutoriales en línea y otros cursos, como el curso de Stanford de Christopher Manning². Las imágenes utilizadas se han extraído intencionalmente directamente de sus fuentes correspondientes. Algún día se diseñarán imágenes propias.

¹El repositorio del curso se encuentra en: <https://github.com/dccuchile/CC6205>

²<http://web.stanford.edu/class/cs224n/>



1. Introducción



2. Modelo de Espacio Vectorial



3. Modelos de Lenguaje Probabilísticos

En esta sección introducimos el concepto de “modelo de lenguaje”. Un enfoque para asignar probabilidades a las oraciones que, como veremos a lo largo del apunte, desempeñará un papel fundamental en los enfoques modernos de PLN.

3.1 El Problema del Modelado del Lenguaje

Supongamos que tenemos un corpus de documentos \mathcal{C} , del cual extraemos su vocabulario (finito) a partir de la tokenización de sus documentos, $\mathcal{V} = \{\text{el, un, hombre, telescopio, Zamorano, dos, fan, vio, jugar, para, Real, Madrid, Zamorano, ...}\}$.

Podemos formar a partir de este vocabulario un conjunto infinito de oraciones si dejamos que éstas sean de largo variable, \mathcal{V}^* .

■ **Ejemplo 3.1** Algunas oraciones que podríamos formar:

- el STOP
- un STOP
- el fan STOP
- el fan vio a Zamorano STOP
- el fan vio vio STOP
- el fan vio a Zamorano jugar por el Real Madrid STOP

Donde STOP es un símbolo especial que indica el final de una oración. ■

Definición 3.1.1 Un modelo de lenguaje es una distribución de probabilidad p sobre todas las oraciones que se pueden formar a partir de un vocabulario finito. Entonces, la probabilidad de cada oración debe ser mayor o igual que cero, y la suma de las probabilidades de todas las oraciones posibles debe ser uno:

$$\sum_{x \in V^*} p(x) = 1$$
$$p(x) \geq 0 \quad \text{para todo } x \in V^*$$

■ **Ejemplo 3.2** A continuación mostramos ejemplos de probabilidades asignadas por un modelo de lenguaje imaginario a algunas oraciones:

$$\begin{aligned} p(\text{el STOP}) &= 10^{-12} \\ p(\text{el fan STOP}) &= 10^{-8} \\ p(\text{el fan vio a Zamorano STOP}) &= 2 \times 10^{-8} \\ p(\text{el fan vio vio STOP}) &= 10^{-15} \\ &\dots \\ p(\text{el fan vio a Zamorano jugar por el Real Madrid STOP}) &= 2 \times 10^{-9} \end{aligned}$$

■

Lo que esperamos del modelo es que le asigne una probabilidad más alta a las oraciones fluidas (aquellas que tienen sentido y son gramaticalmente correctas) de las que no. También esperamos poder estimar esta función de probabilidad a partir del texto (corpus).

A nivel general, el modelo de lenguaje ayuda a los modelos de generación de texto a distinguir entre buenas y malas oraciones.

3.1.1 ¿Por qué queríamos hacer esto?

La motivación original de estos modelos se origina en el problema de reconocimiento del habla o de transcripción automática para ayudar a estos sistemas a discernir entre distintas oraciones compatibles con la señal acústica. Consideremos las siguientes dos oraciones en inglés:

1. “recognize speech” (reconocer el habla)
2. “wreck a nice beach” (arruinar una playa bonita).

Estas dos oraciones suenan muy similares al ser pronunciadas (tienen casi la misma señal acústica), lo que dificulta que los sistemas automáticos de reconocimiento del habla las transcriban con precisión. Cuando el sistema de reconocimiento del habla analiza la entrada de audio e intenta transcribirlo, tiene en cuenta las probabilidades del modelo de lenguaje para determinar la interpretación más probable. El modelo de lenguaje favorecería $p(\text{"recognize speech"})$ sobre $p(\text{"wreck a nice beach"})$. Esto se debe a que la primera oración es más común y debería ocurrir con más frecuencia en el corpus de entrenamiento.

Al incorporar modelos de lenguaje, los sistemas de reconocimiento del habla pueden mejorar la precisión al seleccionar la oración que se alinea mejor con los patrones lingüísticos y el contexto, incluso cuando se enfrentan a alternativas que suenan similar.

De hecho, los modelos de lenguaje son útiles en cualquier tarea de procesamiento del lenguaje natural que involucre la generación de lenguaje (por ejemplo, traducción automática, resumen, generación de resúmenes, chatbots, reconocimiento óptico de caracteres y el reconocimiento de escritura a mano).

Además la noción de modelamiento probabilístico del lenguaje se utiliza en muchos modelos y tareas de PLN, por lo que las técnicas de estimación desarrolladas en este capítulo serán muy útiles para entender muchos problemas en PLN.

3.2 Regla de la Cadena en Modelos de Lenguaje

En el contexto de los modelos de lenguaje, se parte de la premisa de que cualquier oración s puede ser vista como una secuencia de variables aleatorias X_1, X_2, \dots, X_n . Cada una de estas

variables aleatorias puede tomar valores de un conjunto finito V . Para simplificar, asumimos una longitud fija n para la secuencia. Nuestro objetivo consiste en modelar la probabilidad conjunta $p(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$. Es importante notar que utilizamos letras mayúsculas para denotar las variables aleatorias (X_1, X_2, \dots, X_n) y minúsculas para referirnos a instancias específicas de estas variables (x_1, x_2, \dots, x_n), que representan elementos del vocabulario (por ejemplo, “perro”, “gato”, “comida”).

Una forma común de representar secuencias de variables aleatorias es mediante la aplicación de la regla de la cadena de probabilidades que definimos a continuación.

Para un conjunto de variables aleatorias X_1, \dots, X_n , la regla de la cadena de probabilidades nos permite expresar la función de probabilidad conjunta $p(X_1, X_2, \dots, X_n)$ como un producto de n probabilidades condicionales:

$$p(X_1, X_2, \dots, X_n) = p(X_1)p(X_2|X_1)p(X_3|X_2, X_1) \dots p(X_n|X_{n-1}, \dots, X_2, X_1). \quad (3.1)$$

Por ejemplo, en el caso de $n = 3$, la expresión se simplifica a:

$$p(X_1, X_2, X_3) = p(X_1)p(X_2|X_1)p(X_3|X_2, X_1).$$

Al utilizar la definición de probabilidades condicionales, podemos expresar $p(X_2|X_1)$ como $p(X_1, X_2)/p(X_1)$ y $p(X_3|X_2, X_1)$ como $p(X_1, X_2, X_3)/p(X_1, X_2)$. Al sustituir estas expresiones en la cadena de productos, vemos que todas las expresiones de cancelan y finalmente obtenemos $p(X_1, X_2, X_3)$.

La aplicación de la regla de la cadena en los modelos de lenguaje no impone ningún supuesto específico sobre el modelo en sí. Más bien, nos proporciona una forma alternativa de modelar la probabilidad de una oración. Sin embargo, la ventaja de utilizar la regla de la cadena es que nos permite aplicar supuestos y hacer simplificaciones en el modelo para facilitar su implementación y cálculo.

3.3 Mirada Predictiva

La regla de la cadena nos permite darle una interpretación predictiva a los modelos de lenguaje. Uno puede asumir que un modelo de lenguaje una vez entrenado (o construido) a partir de un corpus es capaz de entregar una distribución de probabilidad para todas las palabras a partir de un contexto específico. Supongamos que tenemos un contexto formado por una secuencia de palabras $c = X_{n-1}, \dots, X_1$ y queremos predecir la palabra más probable a partir de ese contexto. Uno podría evaluar la probabilidad condicional $p(X_n|X_{n-1}, \dots, X_1)$ o $p(X_n|c)$ para todas las palabras del vocabulario que puede tomar X_n y escoger la más probable:

$$x_n = \operatorname{argmax}_{X_n \in \gamma} p(X_n|c)$$

■ **Ejemplo 3.3** Supongamos que el contexto es $c = \text{“Cristóbal Colón descubrió”}$. El modelo de lenguaje sería capaz de determinar:

$$p(X|\text{“descubrió”, Colón”, “Cristóbal”})$$

$$\forall X \in V, \quad \text{donde } \sum_{X \in V} p(X|c) = 1$$

Por ejemplo:

$$p(\text{"América"} | \text{"descubrió", "Colón", "Cristóbal"}) = 0,02$$

$$p(\text{"Europa"} | \text{"descubrió", "Colón", "Cristóbal"}) = 0,01$$

$$p(\text{"Chile"} | \text{"descubrió", "Colón", "Cristóbal"}) = 0,001$$

$$p(\text{"camello"} | \text{"descubrió", "Colón", "Cristóbal"}) = 0,00001$$

...

Entonces, nuestro modelo de lenguaje predeciría $X = \text{"América"}$ como la palabra siguiente más probable. ■

En el ejemplo anterior esperamos que un buen modelo de lenguaje asigne una probabilidad mayor a $X = \text{"América"}$ que al resto de las opciones. Para realizar estas predicciones, el modelo de lenguaje necesita capturar las reglas sintácticas y semánticas del lenguaje, así como tener un amplio conocimiento del mundo.

Otra propiedad relevante, que se abordará en el Capítulo 13 sobre grandes modelos de lenguaje, es que se puede codificar prácticamente cualquier tarea de PLN como una instrucción dentro del contexto de un modelo de lenguaje. Por ejemplo, se podría utilizar la instrucción “traduce el siguiente texto de inglés a español: I like dogs”, para luego esperar que el modelo de lenguaje sea capaz de realizar la traducción correcta.

3.4 Mirada Generativa

Los modelos de lenguaje aprendidos a partir de un corpus de texto no solo tienen una capacidad predictiva, sino que también tienen una capacidad generativa. Estos modelos pueden utilizarse para generar oraciones nuevas muestreando secuencialmente a partir de las probabilidades condicionales estimadas.

El proceso generativo implica tomar una palabra inicial y, a medida que avanzamos en la generación, seleccionar sucesivas palabras basadas en las probabilidades condicionales proporcionadas por el modelo de lenguaje. Es similar a extraer bolas (palabras) de una urna, donde los tamaños de las bolas están en proporción a sus frecuencias relativas, que son determinadas por el modelo de lenguaje. Cada vez que se selecciona una palabra, se utiliza como contexto para determinar la siguiente palabra, y así sucesivamente hasta que se haya generado una oración completa.

En este proceso generativo, hay diferentes enfoques que se pueden tomar. Uno de ellos es el muestreo estocástico, donde se elige una palabra de acuerdo con su probabilidad condicional dada por el modelo. Esto permite cierta variabilidad en la generación de oraciones, ya que las palabras menos probables también tienen la posibilidad de ser seleccionadas. Por otro lado, también se puede optar por seleccionar la palabra más probable en cada paso, lo que equivale a predecir la siguiente palabra basándose únicamente en la información disponible hasta ese momento.

3.5 Un Método Ingenuo

Un método muy ingenuo para estimar la probabilidad de una oración es contar todas las apariciones de oraciones idénticas a esta en los datos de entrenamiento y dividirlo por el número total de oraciones de entrenamiento (N).

Más formalmente, sea mi oración $s = x_1, x_2, \dots, x_n$, y $c(x_1, x_2, \dots, x_n)$ el número de veces que ocurre la oración en el corpus de entrenamiento, el modelo de lenguaje ingenuo computa la probabilidad de la siguiente manera:

$$p(x_1, x_2, \dots, x_n) = \frac{c(x_1, x_2, \dots, x_n)}{N}$$

El problema de este enfoque, es que el número de posibles oraciones crece de manera exponencial con su cantidad de palabras y el tamaño del vocabulario. Entonces se vuelve cada vez más improbable que una oración específica haya ocurrido en los datos de entrenamiento. Esto es una consecuencia de la propiedad de **dispersión** del lenguaje planteada en el Capítulo 1.

En consecuencia, muchas oraciones tendrán una probabilidad cero según el modelo ingenuo, lo que lleva a una mala generalización.

3.6 Procesos de Markov

Un proceso de Markov de primer orden asume que la probabilidad de que una variable aleatoria tome un valor depende únicamente del valor inmediatamente anterior en la secuencia. En el contexto del modelado del lenguaje, esto significa que la probabilidad de una palabra en una oración depende solo de la palabra anterior. Si aplicamos este supuesto a la regla de cadena de probabilidades, la probabilidad conjunta de una secuencia de palabras se simplifica en una multiplicación de probabilidades condicionales de las palabras sucesivas dado su palabra predecesora inmediata:

$$p(X_i = x_i | X_1 = x_1, \dots, X_{i-1} = x_{i-1}) = p(X_i = x_i | X_{i-1} = x_{i-1}) \quad (3.2)$$

Un proceso de Markov de segundo orden amplía la suposición de Markov de primer orden y considera el valor de dos variables anteriores en la secuencia. En el modelado del lenguaje, esto significa que la probabilidad de una palabra en una oración depende de las dos palabras anteriores. La probabilidad conjunta de una secuencia de palabras se calcula multiplicando las probabilidades condicionales de las palabras sucesivas dado sus dos palabras predecesoras:

$$p(X_i = x_i | X_1 = x_1, \dots, X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1}) = p(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1}) \quad (3.3)$$

3.6.1 Modelado de secuencias de longitud variable

Si queremos modelar secuencias de longitud variable, podemos considerar que la longitud de la secuencia, n , también es una variable aleatoria. Una forma simple de abordar esto es siempre definir $X_n = \text{STOP}$, donde “STOP” es un símbolo especial que marca el final de la secuencia. Luego, podemos usar un proceso de Markov como antes para modelar la probabilidad conjunta de las palabras en la secuencia:

$$p(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = \prod_{i=1}^n p(X_i = x_i | X_{i-2} = x_{i-2}, X_{i-1} = x_{i-1})$$

Aquí, asumimos que $x_0 = x_{-1} = *$ por conveniencia, donde “*” es un símbolo especial de “inicio” de oración.

3.7 Modelos de lenguaje de Trigramas

Un n-grama es una secuencia contigua de n palabras. Un modelo de lenguaje de trígrama (n-grama con $n = 3$) acoge los supuestos de independencia del modelo de Markov de segundo orden que se discutió anteriormente. De esta forma permite expresar la probabilidad de una oración de la siguiente forma:

$$p(X_1, X_2, \dots, X_n) = p(X_1)p(X_2|X_1)p(X_3|X_2, X_1)\dots p(X_n|X_{n-1}, X_{n-2}). \quad (3.4)$$

Nótese que en vez de condicionar por todas las palabras anteriores como en la Ecuación 3.1 aquí solo se condiciona por las dos palabras anteriores.

Los componentes que definen al modelo de lenguaje de trígrama se definen a continuación:

1. Un vocabulario V , representado por un conjunto finito de palabras.
2. Un parámetro escalar $q(w|u, v)$ para cada trígrama u, v, w , donde $w \in V \cup \{\text{STOP}\}$ y $u, v \in V \cup \{*\}$.

Para cualquier oración $x_1 \dots x_n$, donde $x_i \in V$ para $i = 1 \dots (n - 1)$ y $x_n = \text{STOP}$, la probabilidad de la oración según el modelo de lenguaje trígrama es:

$$p(x_1 \dots x_n) = \prod_{i=1}^n q(x_i|x_{i-2}, x_{i-1})$$

Aquí, definimos $x_0 = x_{-1} = *$ como dos tokens especiales antes del inicio del oración para poder condicionar tanto la primera como la segunda palabra de una oración por sus dos palabras. A este tipo de transformaciones se les conoce en PLN como “padding”.

Por ejemplo, para la oración “el perro ladra STOP”, estimaríamos su probabilidad de la siguiente forma:

$$p(\text{el perro ladra STOP}) = q(\text{el}|*, *) \times q(\text{perro}|*, \text{el}) \times q(\text{ladra}|\text{el}, \text{perro}) \times q(\text{STOP}|\text{perro}, \text{ladra})$$

Ahora abordemos el problema de la estimación de los parámetros $q(w_i|w_{i-2}, w_{i-1})$ a partir de un corpus \mathcal{C} .

Una estimación natural es la “estimación de máxima verosimilitud” (MLE por sus siglas en inglés), donde se estiman los parámetros para maximizar la probabilidad de los datos de entrenamiento.

En el contexto de un modelo de trígrama, la estimación de cada parámetro proviene de distribuciones multinomiales, lo cual se traduce en el siguiente cálculo:

$$q(w_i|w_{i-2}, w_{i-1}) = \frac{\text{Count}(w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-2}, w_{i-1})}$$

Esto requiere tokenizar el corpus de entrenamiento y guardar en una tabla los conteos de todos los trigramas (secuencias contiguas de 3 palabras) y bigramas (secuencias contiguas de dos palabras) encontrados.

Por ejemplo:

$$q(\text{ladra}|\text{el}, \text{perro}) = \frac{\text{Count}(\text{el}, \text{perro}, \text{ladra})}{\text{Count}(\text{el}, \text{perro})}$$

3.8 Evaluación de un modelo de lenguaje: Perplejidad

Un modelo de lenguaje efectivo se caracteriza por asignar probabilidades más altas a oraciones coherentes que a oraciones sin sentido. Para evaluar la calidad de un modelo de lenguaje, es esencial utilizar un corpus de prueba independiente diferente al corpus en el que fue entrenado. Esto nos permitirá verificar cómo generaliza el modelo a nuevas oraciones. La evaluación cuantitativa de modelos de lenguaje nos permite comparar distintos tipos de modelos de lenguaje y seleccionar el modelo más apropiado.

Para evaluar un modelo de lenguaje utilizamos un corpus de prueba independiente, $\mathcal{C}_{\text{test}}$, que contiene m oraciones: $s_1, s_2, s_3, \dots, s_m$, y cuantificamos la probabilidad que el modelo asigna a estas oraciones. Para esto, calculamos la probabilidad total asignada por el modelo a las oraciones del corpus de prueba, lo cual se puede expresar más convenientemente mediante la probabilidad logarítmica¹:

$$\log \left(\prod_{i=1}^m p(s_i) \right) = \sum_{i=1}^m \log p(s_i)$$

La medida de evaluación estándar para modelos de lenguaje es la perplejidad (o perplexity), que se calcula de la siguiente manera:

$$\text{Perplejidad} = 2^{-l} \quad \text{donde} \quad l = \frac{1}{M} \sum_{i=1}^m \log p(s_i)$$

Donde M es el número total de palabras en los datos de prueba.

La perplejidad es una métrica que favorece valores más bajos y tiene la propiedad de ser más fácil de interpretar que la probabilidad logarítmica, como se discutirá a continuación.

3.8.1 Intuición sobre la perplejidad

Supongamos que tenemos un vocabulario V , y $N = |V| + 1$, y un modelo de lenguaje “uniforme” que le asigna la misma probabilidad a todos los trigramas:

$$q(w|u, v) = \frac{1}{N} \quad \text{para todo } w \in V \cup \{\text{STOP}\}, \text{ para todo } u, v \in V \cup \{*\}$$

La perplejidad de este modelo de lenguaje es N tal como se demuestra a continuación.

Supongamos que tenemos m oraciones de longitud n en el corpus, y M es la cantidad de tokens en el corpus, $M = m \cdot n$. Consideremos el logaritmo (base 2) de la probabilidad de una oración $s = w_1 w_2 \dots w_n$ bajo el modelo:

$$\log p(s) = \log \prod_{i=1}^n q(w_i|w_{i-2}, w_{i-1}) = \sum_{i=1}^n \log q(w_i|w_{i-2}, w_{i-1})$$

Dado que cada parámetro $q(w_i|w_{i-2}, w_{i-1})$ es igual a $\frac{1}{N}$, tenemos:

$$\log p(s) = \sum_{i=1}^n \log \frac{1}{N} = n \cdot \log \frac{1}{N} = -n \cdot \log N$$

¹El logaritmo es una función monótona y permite llevar los productos a suma que evita problemas de underflow al trabajar con números pequeños.

$$l = \frac{1}{M} \sum_{i=1}^m \log p(s_i) = \frac{1}{M} \sum_{i=1}^m -n \cdot \log N = \frac{1}{M} \cdot -m \cdot n \cdot \log N = -\log N$$

Por lo tanto, la perplejidad está dada por:

$$\text{Perplejidad} = 2^{-l} = 2^{-(-\log N)} = N$$

Esperamos entonces, que cualquier modelo de lenguaje que sea capaz de aprovechar la distribución de las palabras (como un modelo de trigramas) tenga una perplejidad menor que el tamaño del vocabulario. En [Goodman, 2001], donde $|V| = 50,000$, se presentan los siguientes resultados:

- Modelo de trigramas: $p(x_1, \dots, x_n) = \prod_{i=1}^n q(x_i | x_{i-2}, x_{i-1})$
Perplejidad = 74
- Modelo de bigramas: $p(x_1, \dots, x_n) = \prod_{i=1}^n q(x_i | x_{i-1})$
Perplejidad = 137
- Modelo de unigramas: $p(x_1, \dots, x_n) = \prod_{i=1}^n q(x_i)$
Perplejidad = 955

Estos resultados muestran que la perplejidad disminuye a medida que aumentamos el nivel de contexto considerado en el modelo de lenguaje, lo que indica que los modelos con perplejidad más baja generalmente son más efectivos en la tarea de predicción del lenguaje. Sin embargo, como veremos a continuación, aumentar el tamaño del contexto no está libre de problemas.

3.8.2 El trade-off entre sesgo y varianza

Una limitación de los modelos de trigramas está relacionada con la dispersión del lenguaje, discutida en el Capítulo 1. A grandes rasgos, si el tamaño del vocabulario es $N = |V|$, entonces el modelo tiene N^3 parámetros. Por ejemplo, si $N = 20,000$, habría $20,000^3 = 8 \times 10^{12}$ parámetros. Esta cantidad de parámetros es enorme y dificulta la estimación adecuada de todos ellos sin requerir un corpus de entrenamiento de tamaño enorme. Como resultado, puede surgir un problema de sobreajuste (overfitting) a los datos de entrenamiento. Por otro lado, modelos más simples, como un modelo de lenguaje de unígrafo o de bigrama, no son capaces de modelar adecuadamente el contexto.

Este dilema es conocido en estadística y aprendizaje automático como el trade-off entre sesgo y varianza, que se refiere a la compensación entre la simplicidad del modelo y su capacidad para capturar la complejidad y variabilidad de los datos.

Los modelos más simples, como los modelos de n-gramas de orden inferior (unígramas, bigramas), tienen un sesgo más alto pero una varianza más baja. Estos modelos asumen independencia condicional entre las palabras y simplifican la estructura del lenguaje.

En contraste, los modelos más complejos, como modelos de n-gramas con $n > 3$, tienen una varianza más alta pero un sesgo más bajo. Estos modelos pueden capturar relaciones más complejas entre las palabras, pero también son más propensos a sobreajustar los datos de entrenamiento y tener dificultades para generalizar a nuevas muestras.

Cuando se entrena un modelo de lenguaje con MLE, se maximiza la probabilidad de los datos de entrenamiento. Esto puede llevar a asignar probabilidades altas a secuencias específicas que aparecen en los datos de entrenamiento, incluso si esas secuencias son poco probables en la distribución real del lenguaje.

Como resultado, el modelo puede tener un rendimiento deficiente en datos de prueba que contienen secuencias diferentes a las del conjunto de entrenamiento. Esto se debe a que el modelo se ha sobreajustado a los datos de entrenamiento y ha capturado sus características específicas en lugar de aprender patrones más generales del lenguaje.

Para abordar el problema del sobreajuste en modelos de lenguaje, se utilizan diversas técnicas de regularización. Estas técnicas ayudan a controlar la varianza del modelo y mitigar el sobreajuste, permitiendo un mejor equilibrio entre el sesgo y la varianza y mejorando la generalización a nuevas muestras. A continuación, veremos dos técnicas de regularización para modelos de lenguaje: la interpolación lineal y los modelos de descuento.

3.9 Interpolación Lineal

Se combina la distribución de probabilidad estimada por un modelo de n-gramas con las distribuciones estimadas por modelos de orden inferior. Esto permite incorporar información de n-gramas de orden inferior y reduce la varianza del modelo.

Tomamos nuestra estimación $q(w_i|w_{i-2}, w_{i-1})$ como:

$$q(w_i|w_{i-2}, w_{i-1}) = \lambda_1 \cdot q_{\text{ML}}(w_i|w_{i-2}, w_{i-1}) + \lambda_2 \cdot q_{\text{ML}}(w_i|w_{i-1}) + \lambda_3 \cdot q_{\text{ML}}(w_i)$$

donde $\lambda_1 + \lambda_2 + \lambda_3 = 1$, y $\lambda_i \geq 0$ para todo i .

Nuestra estimación define correctamente una distribución (definimos $V' = V \cup \{\text{STOP}\}$):

$$\begin{aligned} & \sum_{w \in V'} q(w|u, v) \\ &= \sum_{w \in V'} [\lambda_1 \cdot q_{\text{ML}}(w|u, v) + \lambda_2 \cdot q_{\text{ML}}(w|v) + \lambda_3 \cdot q_{\text{ML}}(w)] \\ &= \lambda_1 \sum_w q_{\text{ML}}(w|u, v) + \lambda_2 \sum_w q_{\text{ML}}(w|v) + \lambda_3 \sum_w q_{\text{ML}}(w) \\ &= \lambda_1 + \lambda_2 + \lambda_3 = 1 \end{aligned}$$

También podemos demostrar que $q(w|u, v) \geq 0$ para todo $w \in V'$.

3.9.1 Estimación de los Valores λ

Reservamos parte del conjunto de entrenamiento como datos de *validación*. Definimos $c'(w_1, w_2, w_3)$ como el número de veces que se observa el trigram (w_1, w_2, w_3) en el conjunto de validación. Elegimos $\lambda_1, \lambda_2, \lambda_3$ para maximizar:

$$L(\lambda_1, \lambda_2, \lambda_3) = \sum_{w_1, w_2, w_3} c'(w_1, w_2, w_3) \log q(w_3|w_1, w_2)$$

sujetos a $\lambda_1 + \lambda_2 + \lambda_3 = 1$, y $\lambda_i \geq 0$ para todo i , donde

$$q(w_i|w_{i-2}, w_{i-1}) = \lambda_1 \cdot q_{\text{ML}}(w_i|w_{i-2}, w_{i-1}) + \lambda_2 \cdot q_{\text{ML}}(w_i|w_{i-1}) + \lambda_3 \cdot q_{\text{ML}}(w_i)$$

3.10 Métodos de Descuento

- Consideraremos los siguientes recuentos y estimaciones de máxima verosimilitud:
- Las estimaciones de máxima verosimilitud son altas, especialmente para los elementos con recuentos bajos.

Frase	Recuento	$q_{ML}(w_i w_{i-1})$
the	48	
the, dog	15	15/48
the, woman	11	11/48
the, man	10	10/48
the, park	5	5/48
the, job	2	2/48
the, telescope	1	1/48
the, manual	1	1/48
the, afternoon	1	1/48
the, country	1	1/48
the, street	1	1/48

- Definimos los recuentos "descontados" de la siguiente manera:

$$\text{Recuento}^*(x) = \text{Recuento}(x) - 0,5$$

Frase	Recuento	Recuento [*] (x)	$q_{ML}(w_i w_{i-1})$
the	48		
the, dog	15	14.5	14,5/48
the, woman	11	10.5	10,5/48
the, man	10	9.5	9,5/48
the, park	5	4.5	4,5/48
the, job	2	1.5	1,5/48
the, telescope	1	0.5	0,5/48
the, manual	1	0.5	0,5/48
the, afternoon	1	0.5	0,5/48
the, country	1	0.5	0,5/48
the, street	1	0.5	0,5/48

- Las nuevas estimaciones se basan en los recuentos descontados.
- Ahora tenemos cierta "masa de probabilidad faltante":

$$\alpha(w_{i-1}) = 1 - \sum_w \frac{\text{Recuento}^*(w_{i-1}, w)}{\text{Recuento}(w_{i-1})}$$

Por ejemplo, en nuestro caso:

$$\alpha(\text{the}) = \frac{10 \times 0,5}{48} = \frac{5}{48}$$

3.10.1 Modelos de Katz Back-Off (Bigramas)

- Para un modelo de bigrama, definimos dos conjuntos:

$$A(w_{i-1}) = \{w : \text{Count}(w_{i-1}, w) > 0\}$$

$$B(w_{i-1}) = \{w : \text{Count}(w_{i-1}, w) = 0\}$$

- Un modelo de bigrama:

$$q_{\text{BO}}(w_i|w_{i-1}) = \begin{cases} \frac{\text{Count}^*(w_{i-1}, w_i)}{\text{Count}(w_{i-1})} & \text{si } w_i \in A(w_{i-1}) \\ \frac{\alpha(w_{i-1}) q_{\text{ML}}(w_i)}{\sum_{w \in B(w_{i-1})} q_{\text{ML}}(w)} & \text{si } w_i \in B(w_{i-1}) \end{cases}$$

- Donde:

$$\alpha(w_{i-1}) = 1 - \sum_{w \in A(w_{i-1})} \frac{\text{Count}^*(w_{i-1}, w)}{\text{Count}(w_{i-1})}$$

- Para un modelo de trigramas, primero definimos dos conjuntos:

$$A(w_{i-2}, w_{i-1}) = \{w : \text{Count}(w_{i-2}, w_{i-1}, w) > 0\}$$

$$B(w_{i-2}, w_{i-1}) = \{w : \text{Count}(w_{i-2}, w_{i-1}, w) = 0\}$$

- Un modelo de trigramas se define en términos del modelo de bigramas:

$$q_{\text{BO}}(w_i|w_{i-2}, w_{i-1}) = \begin{cases} \frac{\text{Count}^*(w_{i-2}, w_{i-1}, w_i)}{\text{Count}(w_{i-2}, w_{i-1})} & \text{si } w_i \in A(w_{i-2}, w_{i-1}) \\ \frac{\alpha(w_{i-2}, w_{i-1}) q_{\text{BO}}(w_i|w_{i-1})}{\sum_{w \in B(w_{i-2}, w_{i-1})} q_{\text{BO}}(w|w_{i-1})} & \text{si } w_i \in B(w_{i-2}, w_{i-1}) \end{cases}$$

- Donde:

$$\alpha(w_{i-2}, w_{i-1}) = 1 - \sum_{w \in A(w_{i-2}, w_{i-1})} \frac{\text{Count}^*(w_{i-2}, w_{i-1}, w)}{\text{Count}(w_{i-2}, w_{i-1})}$$

Los modelos de Katz Back-Off son una técnica utilizada en modelos de lenguaje para abordar el desafío de la escasez de datos. Estos modelos permiten estimar las probabilidades condicionales de palabras en función de contextos más pequeños cuando no hay suficientes datos disponibles para estimar directamente las probabilidades completas.

En un modelo de bigrama, se define el conjunto $A(w_{i-1})$ como el conjunto de palabras w para las cuales la frecuencia de aparición de la secuencia (w_{i-1}, w) es mayor que cero, es decir, $\text{Count}(w_{i-1}, w) > 0$. Por otro lado, el conjunto $B(w_{i-1})$ se define como el conjunto de palabras w para las cuales la frecuencia de aparición de la secuencia (w_{i-1}, w) es igual a cero, es decir, $\text{Count}(w_{i-1}, w) = 0$.

En un modelo de bigrama, la probabilidad condicional $q_{\text{BO}}(w_i|w_{i-1})$ se calcula de la siguiente manera: si la palabra w_i está en el conjunto $A(w_{i-1})$, se utiliza una estimación basada en las frecuencias relativas de la secuencia (w_{i-1}, w_i) dividida por la frecuencia de w_{i-1} . Por otro lado, si w_i está en el conjunto $B(w_{i-1})$, se utiliza una estimación suavizada que combina una constante de suavizado $\alpha(w_{i-1})$ y las probabilidades condicionales de máxima verosimilitud $q_{\text{ML}}(w_i)$ de las palabras en el conjunto $B(w_{i-1})$.

La constante de suavizado $\alpha(w_{i-1})$ se calcula restando la suma de las frecuencias relativas de las palabras en $A(w_{i-1})$ de uno.

En el caso de un modelo de trigramas, se definen conjuntos similares $A(w_{i-2}, w_{i-1})$ y $B(w_{i-2}, w_{i-1})$ para las secuencias trigramas. La probabilidad condicional $q_{\text{BO}}(w_i|w_{i-2}, w_{i-1})$ en un modelo de trigramas se calcula utilizando el modelo de bigrama correspondiente y aplicando la misma lógica de suavizado basado en los conjuntos $A(w_{i-2}, w_{i-1})$ y $B(w_{i-2}, w_{i-1})$.

Estos modelos de Katz Back-Off permiten aproximar las probabilidades condicionales en situaciones donde la información disponible es limitada, al aprovechar información de contextos más pequeños cuando no se dispone de suficientes datos para estimaciones directas.

3.11 Historia

En 1951, mientras trabajaba en los laboratorios Bell, Claude Shannon (en la Figura 3.1) realizó experimentos sobre la entropía del inglés, modelando el lenguaje escrito de manera estadística y predictiva [Shannon, 1951]. Utilizando modelos de lenguaje de n-gramas, Shannon exploró la dificultad de predecir palabras basándose en las palabras anteriores.

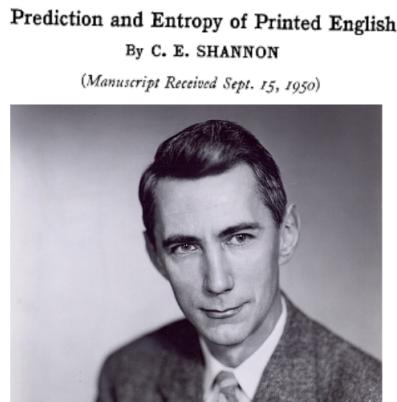


Figura 3.1: Imagen de Shannon.

Por otro lado, en su libro *Syntactic Structures* (1957), Noam Chomsky (en la Figura 3.2), un lingüista y científico cognitivo, cuestionó la capacidad de los modelos de lenguaje probabilísticos para capturar y comprender la gramática del lenguaje humano [Chomsky, 2009]. Según Chomsky, la noción de “gramaticalmente correcto” no puede ser equiparada a “significativo” en un sentido probabilista. Para ilustrar esto, presentó dos oraciones ficticias, ambas carentes de sentido:

1. Colorless green ideas sleep furiously.
2. Furiously sleep ideas green colorless.

Aunque ambas oraciones carecen de significado, Chomsky argumentó que solo la primera se considera gramaticalmente correcta por los hablantes de inglés. Además, enfatizó que la corección gramatical en inglés no puede determinarse únicamente mediante aproximaciones estadísticas. Aunque es poco probable que ninguna de las dos oraciones (1) o (2) haya surgido en documentos escritos en inglés, un modelo estadístico como los modelos de lenguaje vistos en este capítulo las consideraría igualmente “remotas” en relación al inglés. Sin embargo, la oración (1) es gramaticalmente correcta, mientras que la oración (2) no lo es, lo que destaca las limitaciones de los enfoques estadísticos para capturar la gramática. Estos argumentos retrasaron el estudio de los modelos de lenguaje probabilísticos durante varios años [Jurafsky and Martin, 2023].

3.12 Conclusiones

La derivación de probabilidades en modelos de lenguaje probabilísticos implica tres pasos:

1. Expandir $p(w_1, w_2, \dots, w_n)$ usando la regla de la Cadena.
2. Aplicar los supuestos Independencia de Markov

$$p(w_i|w_1, w_2, \dots, w_{i-2}, w_{i-1}) = p(w_i|w_{i-2}, w_{i-1}).$$
3. Suavizar las estimaciones utilizando conteos de orden inferior.

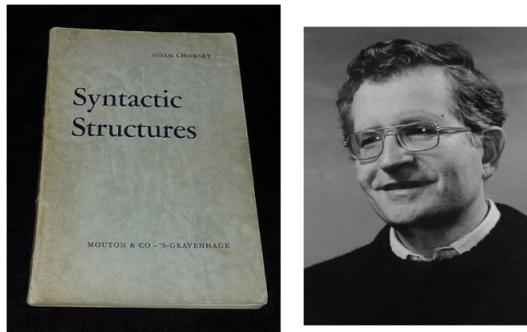


Figura 3.2: Imagen de Chomsky

No obstante, los modelos de lenguaje de bigramas o trigramas (que consideran dos palabras anteriores como contexto) tienen limitaciones en contextos largos y no pueden aprovechar contextos similares. Por ejemplo, consideremos los contextos:

- c_1 : Despues de comer cereales
- c_2 : Luego de desayunar avena

Aunque esperaríamos que las distribuciones de probabilidad $p(w|c_1)$ y $p(w|c_2)$ fueran similares, dado que c_1 y c_2 casi no comparten palabras, los modelos de n-gramas que se limitan a contar frecuencia de palabras no pueden capturar estas similitudes entre contextos.

Otros métodos para mejorar los modelos de lenguaje incluyen introducir variables latentes para representar tópicos, conocidos como modelos de tópicos [Blei et al., 2003] presentados en el Capítulo 2. O alternativamente, reemplazar $p(w_i|w_1, w_2, \dots, w_{i-2}, w_{i-1})$ con una red neuronal predictiva y una “capa de embedding” para representar mejor contextos más grandes y aprovechar similitudes entre palabras en el contexto. [Bengio et al., 2000]

Los modelos de lenguaje modernos utilizan redes neuronales profundas en su estructura principal y tienen un vasto espacio de parámetros como se verá en el Capítulo 13.



4. Clasificación de Texto y Naïve Bayes



5. Modelos Lineales



6. Redes Neuronales



7. Vectores de Palabra



8. Etiquetado de Secuencias

9. Redes Neuronales Convolucionales



10. Redes Neuronales Recurrentes



11. Modelos Secuencia a Secuencia



12. Arquitectura de Transformer



13. Grandes Modelos de Lenguaje



Bibliografía

- [Amir et al., 2015] Amir, S., Ling, W., Astudillo, R., Martins, B., Silva, M. J., and Trancoso, I. (2015). Inesc-id: A regression model for large scale twitter sentiment lexicon induction. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 613–618, Denver, Colorado. Association for Computational Linguistics.
- [Baroni et al., 2014] Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 238–247. Association for Computational Linguistics.
- [Bender, 2013] Bender, E. M. (2013). Linguistic fundamentals for natural language processing: 100 essentials from morphology and syntax. *Synthesis lectures on human language technologies*, 6(3):1–184.
- [Bender et al., 2021] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.
- [Bengio, 2012] Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer.
- [Bengio et al., 2000] Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, 13.
- [Bikel et al., 1999] Bikel, D. M., Schwartz, R. M., and Weischedel, R. M. (1999). An algorithm that learns what’s in a name. *Mach. Learn.*, 34(1-3):211–231.
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.
- [Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

- [Bojanowski et al., 2016] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- [Brown et al., 1992] Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.
- [Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- [Cho, 2015] Cho, K. (2015). Natural language understanding with distributed representation. *arXiv preprint arXiv:1511.07916*.
- [Cho et al., 2015] Cho, K., Courville, A., and Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. *IEEE Transactions on Multimedia*, 17(11):1875–1886.
- [Cho et al., 2014] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- [Chomsky, 2009] Chomsky, N. (2009). Syntactic structures. In *Syntactic Structures*. De Gruyter Mouton.
- [Chung et al., 2022] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling instruction-finetuned language models.
- [Chung et al., 2014] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
- [Collins, 2013] Collins, M. (2013). Statistical nlp lecture notes. *Lecture Notes*.
- [Collobert et al., 2011] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.
- [Conneau et al., 2017] Conneau, A., Schwenk, H., Barrault, L., and Lecun, Y. (2017). Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116.
- [Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.
- [Deng and Liu, 2018] Deng, L. and Liu, Y. (2018). *Deep Learning in Natural Language Processing*. Springer.

- [Duchi et al., 2011] Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.
- [Eisenstein, 2018] Eisenstein, J. (2018). Natural language processing. Technical report, Georgia Tech.
- [Felbo et al., 2017] Felbo, B., Mislove, A., Søgaard, A., Rahwan, I., and Lehmann, S. (2017). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 1615–1625.
- [Fort, 2016] Fort, K. (2016). *Collaborative annotation for reliable natural language processing: Technical and sociological aspects*. John Wiley & Sons.
- [Fromkin et al., 2018] Fromkin, V., Rodman, R., and Hyams, N. (2018). *An introduction to language*. Cengage Learning.
- [Glorot et al., 2011] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323.
- [Goldberg, 2016] Goldberg, Y. (2016). A primer on neural network models for natural language processing. *J. Artif. Intell. Res.(JAIR)*, 57:345–420.
- [Goldberg, 2017] Goldberg, Y. (2017). Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 10(1):1–309.
- [Goldberg and Levy, 2014] Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.
- [Goodman, 2001] Goodman, J. T. (2001). A bit of progress in language modeling. *Computer Speech & Language*, 15(4):403–434.
- [Harris, 1954] Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366.
- [Howard and Ruder, 2018] Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

- [Huang et al., 2015] Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.
- [Johnson, 2014] Johnson, M. (2014). Introduction to computational linguistics and natural language processing (slides). 2014 Machine Learning Summer School.
- [Jozefowicz et al., 2015] Jozefowicz, R., Zaremba, W., and Sutskever, I. (2015). An empirical exploration of recurrent network architectures. In *International conference on machine learning*, pages 2342–2350.
- [Jurafsky and Martin, 2023] Jurafsky, D. and Martin, J. H. (2023). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, NJ, USA, 3rd (draft) edition.
- [Kenton and Toutanova, 2019] Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- [Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- [Kupiec, 1992] Kupiec, J. (1992). Robust part-of-speech tagging using a hidden markov model. *Computer speech & language*, 6(3):225–242.
- [Lafferty et al., 2001] Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Brodley, C. E. and Danyluk, A. P., editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.
- [LeCun et al., 1998] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- [Levy and Goldberg, 2014] Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185.
- [Li et al., 2015] Li, J., Luong, M.-T., Jurafsky, D., and Hovy, E. (2015). When are tree structures necessary for deep learning of representations? *arXiv preprint arXiv:1503.00185*.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- [McCallum et al., 2000] McCallum, A., Freitag, D., and Pereira, F. C. (2000). Maximum entropy markov models for information extraction and segmentation. In *Icml*, volume 17, pages 591–598.
- [McCallum et al., 1998] McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI.

- [Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.
- [Mohammad et al., 2013] Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*.
- [Nakov et al., 2013] Nakov, P., Rosenthal, S., Kozareva, Z., Stoyanov, V., Ritter, A., and Wilson, T. (2013). Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises*, pages 312–320, Atlanta, Georgia, USA. Association for Computational Linguistics.
- [Nesterov, 2018] Nesterov, Y. (2018). *Lectures on convex optimization*, volume 137. Springer.
- [OpenAI, 2023] OpenAI (2023). Gpt-4 technical report.
- [Ouyang et al., 2022] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- [Page et al., 1998] Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The pagerank citation ranking: Bring order to the web. Technical report, Technical report, stanford University.
- [Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar; A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.
- [Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- [Polyak, 1964] Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17.
- [Pustejovsky and Stubbs, 2012] Pustejovsky, J. and Stubbs, A. (2012). *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. .”Reilly Media, Inc.”.
- [Radford et al., 2018] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training.
- [Ramshaw and Marcus, 1999] Ramshaw, L. A. and Marcus, M. P. (1999). *Text Chunking Using Transformation-Based Learning*, pages 157–176. Springer Netherlands, Dordrecht.

- [Read, 2005] Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, ACLstudent '05, pages 43–48, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Salton et al., 1975] Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.
- [Severyn and Moschitti, 2015] Severyn, A. and Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962, New York, NY, USA. ACM.
- [Shannon, 1951] Shannon, C. E. (1951). Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.
- [Snow et al., 2008] Snow, R., O’connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263.
- [Srivastava et al., 2014] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.
- [Sutskever et al., 2013] Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147.
- [Tang et al., 2015] Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.
- [Tang et al., 2014] Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014). Building large-scale twitter-specific sentiment lexicon : A representation learning approach. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pages 172–182.
- [Thoppilan et al., 2022] Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. (2022). Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- [Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.
- [Tieleman and Hinton, 2012] Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSEERA: Neural networks for machine learning*, 4(2):26–31.
- [Turian et al., 2010] Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.

- [Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [Wei et al., 2022] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., and Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- [Yule, 2016] Yule, G. (2016). *The study of language*. Cambridge university press.
- [Zeiler, 2012] Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- [Zhao et al., 2023] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A survey of large language models.
- [Zipf, 1935] Zipf, G. K. (1935). *The Psychobiology of Language*. Houghton-Mifflin, New York, NY, USA.
- [Zou and Hastie, 2005] Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.