EL MODELO DE N-GRAMAS Y SUS APLICACIONES

YURY VIVIANA HOYOS
SANDRA GIMENA MARTINEZ

UNIVERSIDAD DEL CAUCA
FACULTAD, CIENCIAS NATURALES EXACTAS Y DE LA EDUCACIÓN
DEPARTAMENTO DE MATEMATICAS
POPAYAN,CAUCA
AGOSTO, 2010

EL MODELO DE N-GRAMAS Y SUS APLICACIONES

Yury Viviana Hoyos
Sandra Gimena Martinez

Trabajo de grado presentado en forma de seminario y dirigido por el
Dr. Fredy Amaya para optar el título de Matemático.

Universidad del Cauca
Facultad, Ciencias naturales exactas y de la educación
Departamento de Matemáticas
Popayán,Cauca
Agosto, 2010

NOTA DE ACEPTACIÓN

Director

Doctor. Fredy Angel Amaya Robayo

JURADOS

profesor. Diego Fernando Ruiz

profesor. Diego Correa

AGRADECIMIENTOS

Damos gracias a Dios por su infinita ayuda y por la maravillosa oportunidad de ser profesionales, a nuestros padres por caminar una vez más con nosotras en esta etapa de nuestra vida.

Agradecemos a nuestro director de tesis el doctor Freddy Amaya por sus aportes y enseñanzas durante este proceso, a los miembros del comité de seguimiento los profesores Diego Correa y Diego Ruiz por su acompañamiento y dedicación.

A nuestros amigos y compañeros con los que compartimos experiencias, conocimientos y que de una u otra forma aportaron para alcanzar esta triunfo.

Indice general

INTRODUCCIÓN

Los modelos del lenguaje son una tarea central en los sistemas de reconocimiento automático del habla y otras aplicaciones. Un modelo del lenguaje es un modelo matemático encargado de asignar probabilidades a un lenguaje, que en su forma más simple puede ser una representación de las palabras que pertenecen al lenguaje y en los modelos más complejos también se trata de describir la estructura y significado de las frases pertenecientes al lenguaje. Se encuentran varios tipos de modelos de lenguaje adecuados a cada una de las diferentes aplicaciones del reconocimiento de formas.

El propósito de este trabajo es estudiar el modelo de n-gramas, el cual es el modelo de lenguaje que ha demostrado mayor efciencia y sencillez en su implementación; esto hace que el modelo sea el más utilizado en los sistemas de reconocimiento del habla como también en otros campos en los que se procesa información lingüística mediante sistemas probabilísticos. Por otra parte, éste modelo presenta problemas de dispersión, pero existe una técnica utilizada para corregir este problema, la cual es el suavizado. El suavizado es la técnica mediante la cual se ajustan las probabilidades estimadas en el modelo. El uso de ésta técnica produce distribuciones más uniformes, aumentando así la capacidad predictiva de los modelos.

El contenido de éste trabajo se divide en cinco capítulos, en el primero se realiza una revisión acerca de los modelos de lenguaje. En el segundo se expone la teoría de los modelos de n-gramas y de los algoritmos de suavizados para estos modelos. En el capítulo tres se estudian los modelos mixtos, los cuales son combinaciones del modelo de n-gramas con otros modelos mediante algún método. El capítulo cuatro muestra la descripción y resultados de los experimentos. Y por último, en el capítulo cinco se mencionan las conclusiones y se sugieren trabajos futuros.

Capítulo 1

Modelos de lenguaje

1.1. Definición y utilidad de los modelos de lenguaje

El reconocimiento de patrones o reconocimiento de formas es un importante campo de la informática encargado de diseñar e implementar sistemas automáticos capaces de reconocer objetos de diferente naturaleza.

Si consideramos al cerebro como una máquina y a sus órganos asociados (ej. el oído) como traductores que proveen un canal de entrada, que reciben una señal del mundo externo y la envían al cerebro para su análisis y reconocimiento, podría pensarse que un proceso similar pueda realizarse mediante un sistema de cómputo, de tal forma que éste sea capaz de reconocer objetos o señales.

Sin embargo, es muy poco real pensar que un modelo de reconocimiento de patrones funcione tan sofisticadamente como el cerebro humano, aunque no se niegan los avances obtenidos en esta área en los últimos años.

Entre las aplicaciones del reconocimiento de patrones, están el reconocimiento de la voz, el reconocimiento de caracteres manuscritos, el reconocimiento de las huellas dactilares, el reconocimiento del iris y muchas más.
Para ilustrar cómo funcionan los sistemas de reconocimiento automático de patrones, se toma el caso del problema del reconocimiento de la voz. Ver Figura 1.

Fig. 1 Proceso de reconocimiento de la voz

Una persona genera una onda acústica, ésta es capturada por un micrófono que la lleva hasta un sistema de digitalización, el cual se encarga de asignar símbolos que identifiquen la onda acústica dependiendo de las características de la voz como la amplitud, el timbre, el color, etc. Una vez asignados estos símbolos pasan a la etapa de preproceso, de donde se obtiene un vector $O = o_1 o_2 ... o_n$; donde o_i, i = 1,...,n son vectores numéricos asignados por el sistema de preproceso y entran a un sistema de reconocimiento que se encarga de asignar probabilidades a las secuencias $w = w_1, \ldots, w_n$ que pertenece a W^+, siendo W^+ el conjunto de todas las cadenas formadas con elementos del vocabulario W. El problema que trata el sistema de reconocimiento es encontrar la frase $\hat{w}$ en W^+ que tiene la mayor probabilidad de ocurrir dada la información O, es decir:

$$\hat{w} = Arg\left(\max_{w \in W^+} p(w \mid O)\right)$$

Haciendo uso del teorema de Bayes y dado que $O \notin W^+$, $p(O)$ no influye en

el resultado de $\hat{w}$, se tiene entonces que

$$\hat{w} = Arg\left(\max_{w \in W^+} p(O \mid w)p(w)\right)$$

A $p(O \mid w)$ se le denomina Modelo Acústico y se encarga de determinar el grado de probabilidad de que la secuencia acústica generada corresponda a la frase w. Una alta probabilidad de $p(O \mid w)$ significa que dada la evidencia de w, es bastante posible que la secuencia acústica O corresponda a dicha frase. Por el contrario, si $p(O \mid w)$ tiene un valor bajo, lo más probable es que la secuencia acústica no corresponda a la frase dada. La probabilidad $p(w)$ se conoce como Modelo de lenguaje, porque está encargada de modelar la probabilidad de ocurrencia de la frase w en el lenguaje. Tanto al modelo acústico como al modelo de lenguaje por separado se les ha dedicado una amplia investigación dando lugar al desarrollo de gran cantidad de modelos de lenguaje.

La construcción de Modelos del lenguaje es una tarea central en los sistemas de reconocimiento automático del habla, procesamiento del lenguaje natural, reconocimiento del iris, reconocimiento de huellas dactilares, traducción automática entre otras aplicaciones. Para cada una de estas aplicaciones es necesario contar con un conjunto de elementos llamado vocabulario. Un vocabulario es un conjunto de términos elementales que forman parte de un lenguaje específico; por lenguaje se entiende un sistema de códigos que ayudan a distinguir acciones, cualidades y relaciones entre objetos del mundo exterior.

Los lenguajes formales son construcciones artificiales humanas que se usan en matemáticas y otras disciplinas formales, incluyendo lenguajes de programación. Estas construcciones tienen estructuras internas que comparten con el lenguaje humano natural, por lo que pueden ser en parte analizados con los mismos conceptos de nuestro lenguaje. En el caso particular del lenguaje natural, éste está formado por un grupo de frases; en la que una frase es una

secuencia de palabras de un conjunto llamado vocabulario.

Un modelo de lenguaje es un modelo matemático que se encarga de asignar probabilidades a una secuencia de palabras, es decir, le asigna probabilidades a un lenguaje. En su forma más simple puede ser una representación de las palabras que pertenecen al lenguaje, y en los modelos más complejos también se describe la estructura y significado de las frases pertenecientes al lenguaje.
Dado que el modelo de lenguaje asigna probablidades a una frase o cadena w, éste muestra qué tan frecuentemente ocurre w como oración; probabilidades bajas significarán que la frase no es usual en el lenguaje o no pertenece a él, mientras que probabilidades altas significan que la frase ocurre con mucha frecuencia; por ejemplo $p(w = hola) \approx 0{,}01$ significa tal vez que de cada 100 oraciones que una persona habla, pronuncia una vez la frase $w =$ hola. De esa manera, el modelo del lenguaje ayuda al sistema a reducir la búsqueda descartando aquellas frases que tienen muy poca probabilidad de ocurrir.

Si $w = w_1 w_2 ... w_n$ es una frase, entonces utilizando el teorema de Bayes la probabilidad $p(w)$ se puede expresar como:

$$p(w) = \prod_{i=1}^{n} p(w_i \mid w_1 ... w_{i-1})$$

A la secuencia $w_1 w_2 ... w_{i-1}$, se le denomina historia de w_i y se denota h_i. Al conjunto de las probabilidades $\{p(w_i \mid w_1 ... w_{i-1})\}$ para $i = 1, \ldots, n$ se le denomina Modelo Condicional. Puede observarse que la estimación de $p(w)$ no es práctica cuando $|W|$ es grande ($|W| \equiv$ cardinal de W, número de elementos del conjunto), pues para calcular $p(w_i \mid w_1 ... w_{i-1})$ debemos estimar la probabilidad de $|W|^i$ eventos.

1.2. Tipos de modelos de lenguaje

A continuación se mencionarán diferentes tipos de modelos de lenguaje adecuadas a las diferentes aplicaciones en el reconocimientos de formas.

1.2.1. Modelos basados en gramáticas

Los modelos de lenguaje basados en gramáticas representan las restricciones del lenguaje de una manera natural y permiten modelar dependencias tan largas como se quiera.

Se mencionarán algunas ventajas de los modelos de lenguaje basados en gramáticas:

- Para estos modelos resultan particularmente interesantes los casos en los que el lenguaje tratado es especializado. Es el caso de las aplicaciones destinadas al reconocimiento de voz, para la búsqueda de informaciones vinculadas a los pasajes de avión y pasar de voz a texto o al reconocimiento de la escritura manuscrita, por ejemplo, el escrito sobre los cheques de banco y reconocedor de texto o letras.

- El lenguaje empleado en este tipo de aplicaciones es muy reducido con respecto al lenguaje natural y por lo tanto mucho más fácil de modelar.

Desventaja de los modelos de lenguaje basados en gramáticas:

- Estos modelos tratan una frase en su totalidad, lo cual dificulta su adaptación a un reconocimiento en tiempo real, en el que solamente se dispone de la historia de la palabra actual.

1.2.2. Modelos estadísticos

Los modelos de lenguaje estadísticos se estiman generalmente a partir de un corpus (definido en el siguiente párrafo) delimitándolos por el tamaño del vocabulario, la longitud del contexto e incluyendo esquemas para tratamiento de palabras desconocidas. Uno de los factores determinantes de un modelo de lenguaje es el tamaño del corpus usado durante la fase de entrenamiento, mientras más grande sea el corpus mayor será el número de contextos de uso de una palabra dada, esto hace que los modelos de lenguaje estadísticos tengan mayor rendimiento, eficiencia y sean más flexibles que los basados en gramáticas.

Por Corpus se entiende grandes cantidades de datos que son seleccionados y ordenados según criterios lingüísticos explícitos con el fin de ser utilizados como una muestra representativa de la lengua. Un corpus debe estar compuesto por textos producidos en situaciones reales llamados datos y la inclusión de los textos que componen el corpus debe estar guiada por una serie de criterios lingüísticos.

Veamos ahora algunas de las ventajas que presenta un modelo estadístico:

- No es necesario contar con la intervención de expertos en el idioma o dominio a tratar.

- La transposición sobre nuevos datos. Cuando el proceso de aprendizaje se automatiza completamente es posible incorporar nuevos datos para el aprendizaje de tal forma que se mejora la solidez de las estimaciones probabilistas o se especializa el modelo sobre una base específica a un dominio.

- Una estimación probabilista es más fina y más flexible que una esti-

mación que solamente puede determinar si acepta o rechaza una secuencia de palabras, puesto que la estimación probabilista asigna valores reales de la ocurrencia de una frase en los datos de entrenamiento.

- El tiempo de cálculo empleado es más reducido.

- Permite tratar los fenómenos lingüísticos por una observación de eventos elementales, tales como la sucesión de palabras, sin necesidad de abordar un enfoque lingüístico más profundo.

Algunas desventajas de este modelo son:

- Los modelos probabilistas no permiten capturar el sentido de una frase. Es posible en este contexto asignar una viabilidad importante a una frase absurda desde el punto de vista semántico.

- Necesita disponer de grandes cantidades de datos de aprendizaje para construir modelos de lenguaje más sólidos.

- La dependencia de la fuente de información. Los conceptos interiores del modelo estadístico no son de fácil comprensión para las personas, lo cual dificulta la posibilidad de integrar manualmente correcciones y mejoras.

1.2.3. Modelos basados en grafos

Los modelos de lenguaje representados mediante grafos [2] heredan las ventajas de las propiedades características de estos, y al representar el modelo por medio de un grafo, el cálculo de las probabilidades $p(w)$ no depende

del modelo de lenguaje particular, se deriva de las probabilidades de la secuencia de nodos recorridos. Los modelos basados en grafos también tienen sus desventajas, principalmente el coste computacional, tanto en el tiempo que se requiere para generar las probabilidades de una secuencia de palabras, como en el espacio necesario para almacenar la estructura del grafo. Dentro de los modelos basados en grafos se tienen los siguientes:

Modelos basados en árboles de decisión. Una técnica de clasificación y a la vez un método de construcción de modelos del lenguaje es la basada en árboles de decisión. Un árbol de decisión es un árbol binario, cuyos nodos se van generando de acuerdo con la respuesta de tipo binario a ciertas preguntas sobre la historia de la palabra a procesar. Específicamente, si h_i es la historia de la palabra w_i, en cada nodo no terminal se lanza una pregunta relativa a la historia h_i que requiere una respuesta binaria (si/no); la respuesta dada conduce a otro nodo donde otra pregunta del mismo estilo se lleva a cabo; el proceso continúa hasta llegar a los nodos terminales (hojas) del árbol. Asociada con cada hoja del árbol hay una clase $\Phi(h_i)$ [1] donde cada historia es finalmente clasificada. Una vez que se tienen clasificadas las historias se procede a estimar las probabilidades $p(w_i|\Phi(h_i))$. Obsérvese que el modelo de n-gramas [2] es un caso particular de éste modelo, las clases de n-gramas se pueden generar si las preguntas en cada nodo deciden sobre la pertenencia o no de las $n-1$ anteriores palabras a la historia h_i. De lo anterior se infiere que un modelo construido óptimamente mediante árboles de decisión es al menos tan bueno como un modelo de n-gramas. Resultados de experimentos reportados en [2] dan cuenta de significativas reducciones en la perplejidad, respecto al modelo de trigramas, cuando éste se combina mediante interpolación lineal con un modelo basado en árbol de decisión.

[1] Es la clase de equivalencia a la que pertenecen todas las cadenas tales que sus historias coinciden en las últimas n - 1 palabras.

[2] Modelo que toma la información aportada por las n - 1 palabras anteriores a n.

Modelos basados en árboles binarios. La característica esencial de estos modelos es la de aprovechar la información aportada por la estructura jerárquica del lenguaje natural contenida en cada frase, de tal forma que de la historia de la palabra a predecir se consideren solamente aquellas palabras que sean relevantes dentro de la estructura sintáctica de la frase. El modelo va construyendo la estructura del árbol incrementalmente a medida que recorre cada frase de izquierda a derecha.
Cada hoja del árbol es una palabra, los nodos están etiquetados (las etiquetas se denominan encabezados) y la etiqueta de cada nodo le da la orientación (así el árbol es un grafo orientado). Las etiquetas están divididas en dos clases: POS para las hojas del árbol y NT, las etiquetas para los nodos no terminales. Dada la cadena w de longitud n, ésta es precedida por el símbolo $< s >$ y finalizada con el símbolo $< /s >$, de manera que se tiene $w_0 =< s >$, $w_{n+1} =< /s >$.
Sea $W_k = w_0 \dots w_k$ una subcadena y $W_k T_k$ el árbol de derivación correspondiente, éste se denomina árbol de derivación k-ésimo y contiene solamente los subárboles cuya expansión esté contenida en W_k excluyendo w_0. Las palabras aisladas con su etiqueta POS, se consideran subárboles que solamente poseen raíz.

1.2.4. Modelos basados en máxima entropía

El concepto básico de entropía en teoría de la información tiene mucho que ver con la incertidumbre que existe en cualquier experimento o señal aleatoria. Es también la cantidad de ruido o desorden que contiene o libera un sistema. De esta forma, podremos hablar de la cantidad de información que lleva una señal.

El principio de Máxima Entropía. El principio de Máxima Entropía es un marco bastante útil y flexible para el desarrollo de modelos del leguaje. Bajo este marco se pueden cobijar unidades lingüísticas de diferente natu-

raleza (palabras, clases, etc.), asi como modelos lingüístistos diversos (*n-gramas*, *modelos con caché*, *gramáticas*, *triggers* etc.), todos unificados por un mismo formalismo. Este principio trata de determinar la distribución probabilística p que haga máxima la entropía, notada por $H(p)$, haciendo uso sólamente de la información de la que se dispone, sin hacer suposiciones teóricas sobre la forma u otra característica de la distribución.

Sea $\mathbf{x} = x_1, \ldots, x_n$ un vector aleatorio n dimensional y $k_i(\mathbf{x})$, $i = 1, \ldots, m$, funciones de restricción. Se plantea encontrar la función de probabilidad $p(\mathbf{x})$ tal que satisface la restriccion:

$$\sum_{\mathbf{x}} p(\mathbf{x}) k_i(\mathbf{x}) = d_i \tag{1.1}$$

para ciertas funciones objetivo d_i $i = 1, 2, \ldots, m$.

Si la función $k_i(\mathbf{x})$ es una función indicatriz, para asegurar que p sea realmente una función de probabilidad se agrega una restricción adicional, dígase la *0-ésima* restricción, $k_0(\mathbf{x}) = 1$ para todo $\mathbf{x}$ y $d_0 = 1$.

Un modelo basado en el principio de máxima entropía

La mayor virtud del principio de máxima entrópia, en lo que al modelado del lenguaje se refiere, es que bajo su marco se pueden combinar estadísticas de datos provenientes de fuentes de información de la más variada naturaleza. Por ejemplo, en [3] el modelo se basa en el concepto y características de *trigger*, que captura información a larga distancia; pero este modelo también se puede definir mediante las características del modelo de n-gramas.

Triggers

Si una secuencia A de palabras está fuertemente correlacionada con otra B, la pareja $(A \rightarrow B)$ se denomina *trigger*, donde A es la secuencia *activadora* y B la *activada*. Por ejemplo, cuando A aparece en el documento *activa* a B causando que cambie la estimación de su probabilidad. En este caso la

función de restricción definida anteriormente será

$$k_{(A\to B)}(w) = \begin{cases} 1 & si\ (A \to B) \in w \\ 0 & en\ otro\ caso \end{cases}$$

n-gramas

En el caso de unigramas la función k_i se define como:

$$k_{w_1}(h, w) = \begin{cases} 1 & si\ w = w_1 \\ 0 & en\ otro\ caso \end{cases}$$

y $d_{w_1} = \tilde{E}[k_{w_1}] \overset{def.}{=} \frac{1}{n}\sum_{(h,w)\in\Omega} k_{w_1}(h, w)$, siendo Ω el conjunto de entrenamiento. La restricción asociada está por tanto dada mediante la fórmula:

$$\sum_h \tilde{p}(h) \sum_w p(w|h) k_{w_1}(h, w) = \tilde{E}[k_{w_1}]$$

De igual manera para bigramas, las respectivas funciones de restricción y objetivo están dadas por:

$$k_{\{w_1,w_2\}}(h, w) = \begin{cases} 1 & si\ h\ termina\ en\ w_1\ y\ w = w_2 \\ 0 & en\ otro\ caso \end{cases}$$

con restricción asociada:

$$\sum_h \tilde{p}(h) \sum_w p(w|h) k_{\{w_1,w_2\}}(h, w) = \tilde{E}[k_{\{w_1,w_2\}}]$$

Para trigramas se tiene:

$$k_{\{w_1,w_2,w_3\}}(h, w) = \begin{cases} 1 & si\ h\ termina\ en\ (w_1, w_2)\ y\ w = w_3 \\ 0 & en\ otro\ caso \end{cases}$$

con restricción asociada:

$$\sum_h \tilde{p}(h) \sum_w p(w|h) k_{\{w_1,w_2,w_3\}}(h,w) = \tilde{E}[k_{w_1,w_2,w_3}]$$

Triggers: De igual manera como se hizo con los n-gramas se hace con los *triggers*. La función $k_{A\to B}(h,w)$ se define como:

$$k_{A\to B}(h,w) = \begin{cases} 1 & si\ A \in h, w = B \\ 0 & en\ otro\ caso \end{cases}$$

y la respectiva restricción está dada por:

$$\sum_h \tilde{p}(h) \sum_w p(w|h) k_{A\to B}(h,w) = \tilde{E}[k_{A\to B}]$$

Observemos que si además de las restricciones proporcionadas por los trigramas, se adicionan al modelo las proporcionadas por los *triggers*, puede darse el caso de información redundante, es decir, información que es aportada por el modelo de *triggers* que ya se encuentra en el modelo de trigramas; por ejemplo, el *trigger* $(Nueva \to York)$, seguramente tiene una información mutua promedio bastante alta y por tanto es candidato considerable, pero a la vez con seguridad $(Nueva, York)$ es una pareja que tendría alta frecuencia de aparición en el modelo de trigramas.

Capítulo 2

Modelos de N-gramas

En el capítulo anterior se habló sobre la importancia de los modelos de lenguaje en el sistema de reconocimiento de formas, en este capítulo se trata el modelo de n-gramas el cual es el modelo de lenguaje que ha demostrado más eficiencia y sencillez en su implementación.

2.1. Definición y propiedades de los modelos de n-gramas

Sea $w = w_1 w_2 ... w_n$ una secuencia de palabras (una frase) con su correspondiente distribución probabilística *p(w)*, donde $w_i \in W$ y W es un vocabulario. Se puede expresar *p(w)* como:

$$\begin{aligned} p(w) &= p(w_1)p(w_2 \mid w_1)...p(w_n \mid w_1...w_{n-1}) \\ &= \prod_{i=1}^{n} p(w_i \mid w_1...w_{i-1}). \end{aligned} \tag{2.1}$$

El problema del cálculo de *p(w)* se transforma en la estimación de las probabilidades condicionales $\{p(w_i \mid w_1...w_{i-1})\}$ para cada posible secuencia $w_1 w_2 ... w_i$, de elementos de W^+.

El cálculo de la probabilidad $p(w)$ en (2.1) resulta computacionalmente inadecuado aún para vocabularios de tamaño razonable y valores pequeños de i, ya que si $|W| = M$, el número de eventos $w = w_1 \dots w_i$ es M^i. Sin embargo, teniendo en cuenta que es poco probable que la elección de w_i dependa de la historia completa anterior a ella, notada como h_i y definida por $h_i = w_1 \dots w_{i-1}$, se suele dividir la historia h en clases de equivalencia, lo cual reduce el costo computacional. Si $\Phi(w_1, \dots w_{i-1})$ denota la clase de equivalencia de la historia h_i, entonces la probabilidad $p(w)$ se aproximaría mediante la fórmula

$$p(w) \approx \prod_{i=1}^{n} p(w_i | \Phi(w_1 \dots w_{i-1})). \tag{2.2}$$

El arte del *modelado de lenguaje* consiste pues, en determinar de manera apropiada tanto las clases de equivalencia de h_i como el método de estimación de las probabilidades $p(w_i \mid \Phi(w_1 \dots w_{i-1}))$.

La popularidad del modelo de n-gramas está basada principalmente en su sencillez y facilidad de implementación; éste toma la información aportada únicamente por las *n-1* anteriores palabras a w_i, es decir, para estimar la ocurrencia de la palabra w_i solamente se considera la información aportada por las *n-1* palabras más recientes.
Usando el modelo de n-gramas, la relación de equivalencia $\sim$ entre dos cadenas $w_1 \dots w_i$, $w_1^{'} \dots w_i^{'}$ se define como:

$$w \sim w' \Longleftrightarrow w_{i-n+1} \dots w_{i-1} = w_{i-n+1}^{'} \dots w_{i-1}^{'}.$$

Mediante $\Phi(w_1 \dots w_{i-1})$ se nota la clase a la que pertenecen todas las cadenas tales que sus historias coinciden en las últimas $n-1$ palabras.

Para el modelo de n-gramas se pueden considerar distintos casos según el valor que tome n, es decir,

Si n = 1, el modelo se llama **unigrama**
Si n = 2, el modelo se llama **bigrama**
. . .
Si n = i, el modelo se llama **i-grama**.

En la práctica, el n más grande usado es n = 3, este modelo es llamado modelo de trigramas y generalmente da muy buenos resultados.

En adelante y por sencillez, se tratará con el modelo de bigramas, de ahí que el cálculo de $p(w)$ en (2.1), sólo dependerá de la palabra inmediatamente anterior a la palabra a procesar, obteniendo la aproximación:

$$\begin{aligned} p(w) &= p(w_1)p(w_2 \mid w_1)...p(w_n \mid w_1 w_2 ... w_{n-1}) \\ &\approx \prod_{i=1}^{n} p(w_i \mid w_{i-1}). \end{aligned} \tag{2.3}$$

En $w = w_1 w_2 ... w_n$ Chen y Goodman [1] toman los símbolos $\langle \mathbf{BOS} \rangle$ para indicar el comienzo de la frase, es decir, el término w_0, y para el final de la frase toman $\langle \mathbf{eos} \rangle$ que es el término w_{n+1}, esto para garantizar que la suma de todas las probabilidades calculadas sea 1 . Por ejemplo para calcular

$$p(\text{JHON LEE UN LIBRO})$$

usando un modelo de bigrama, en la ecuación (2.3) se tiene:

$$\begin{aligned} p(\text{Jhon lee un libro}) = {} & p(Jhon \mid \langle \mathbf{BOS} \rangle) p(lee \mid Jhon) p(un \mid lee) \\ & p(libro \mid un) p(\langle \mathbf{eos} \rangle \mid libro). \end{aligned}$$

El problema consiste entonces en calcular $p(w_i \mid w_{i-1})$. Como estas probabilidades no se conocen, se deben estimar a partir de una muestra. En el caso de los modelos de lenguaje la muestra debe ser cosistente con el lenguaje que se quiere modelar, por tal razón en vez de muestra aleatoria se utiliza un corpus de datos (ver capítulo 1). La estimación de máxima verosimilitud

para $p(w_i \mid w_{i-1})$ está dada por las frecuencias relativas, es decir,

$$p(w_i \mid w_{i-1}) = \frac{\mathbf{c}(w_{i-1}w_i)}{\sum_{w_j} \mathbf{c}(w_{i-1}w_j)}, \tag{2.4}$$

donde $\mathbf{c}(w_{i-1}w_i)$ denota el número de veces que el bigrama $w_{i-1}w_i$ ocurre en el corpus.

El corpus se divide en dos conjuntos, uno llamado *datos de prueba* (P) y otro *datos de entrenamiento* (E), para los modelos de n-gramas los datos de entrenamiento usan millones de palabras.
Lo dicho anteriormente para el modelo de bigramas se puede generalizar al modelo de n-gramas así:

$$p(w) = \prod_{i=1}^{n+1} p(w_i \mid w_{i-n+1}^{i-1}), \tag{2.5}$$

donde w_i^j denota las palabras $w_i...w_j$. En (2.5), w_{i-n+1}^{i-1} es la clase de equivalencia correspondiente por la relación definida en (2.2). El primer elemento de w_{i-n+1}^{i-1} es generalmente $\langle \mathbf{BOS} \rangle$ y el último elemento w_{n+1} es $\langle \mathbf{eos} \rangle$.

Para estimar las probabilidades $p(w_i \mid w_{i-n+1}^{i-1})$, se hace uso de la ecuación (2.4), con lo que:

$$p(w_i \mid w_{i-n+1}^{i-1}) = \frac{\mathbf{c}(w_{i-n+1}^{i})}{\sum_{w_j} \mathbf{c}(w_{i-n+1}^{i-1}w_j)}, \tag{2.6}$$

siendo $\sum_{w_j} \mathbf{c}(w_{i-n+1}^{i-1}w_j) = \mathbf{c}(w_{i-n+1}^{i-1})$ el conteo de la historia.

A continuación se ilustra lo anterior con un ejemplo. El conjunto de datos de entrenamiento E está compuesto por las frases:

Jhon lee Moby Dick
Mary lee un libro diferente (1)
Ella lee un libro para Cher.

Calculando p(Jhon lee un libro) usando el modelo de bigramas se tiene:

$$p(\text{Jhon lee un libro}) = p(Jhon \mid \langle\mathbf{BOS}\rangle)p(lee \mid Jhon)p(un \mid lee) \\ p(libro \mid un)p(\langle\mathbf{eos}\rangle \mid libro).$$

Aplicando la ecuacion (2.4) en cada uno de los bigramas anteriores,

$$p(Jhon \mid \langle\boldsymbol{BOS}\rangle) = \frac{\mathbf{c}(\langle\mathbf{BOS}\rangle\ Jhon)}{\sum_w \mathbf{c}(\langle\mathbf{BOS}\rangle\ w)} = \frac{1}{3}$$

$$p(lee \mid Jhon) = \frac{\mathbf{c}(Jhon\ lee)}{\sum_w \mathbf{c}(Jhon\ w)} = \frac{1}{1}$$

$$p(un \mid lee) = \frac{\mathbf{c}(lee\ un)}{\sum_w \mathbf{c}(lee\ w)} = \frac{2}{3}$$

$$p(libro \mid un) = \frac{\mathbf{c}(un\ libro)}{\sum_w \mathbf{c}(un\ w)} = \frac{1}{2}$$

$$p(\langle\boldsymbol{eos}\rangle \mid libro) = \frac{\mathbf{c}(libro\ \langle\mathbf{eos}\rangle)}{\sum_w \mathbf{c}(libro\ w)} = \frac{1}{2}$$

obteniendo,

$$p(\text{Jhon lee un libro}) = p(Jhon \mid \langle\boldsymbol{BOS}\rangle)p(lee \mid Jhon)\mathrm{p}(un \mid lee) \\ p(libro \mid un)p(\langle\boldsymbol{eos}\rangle \mid libro).$$

$$= \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2}$$

$$\approx 0{,}06.$$

2.2. Ventajas y Desventajas de los modelos de n-gramas

Ventajas:

A continuación se mencionan algunas ventajas que hacen que el modelo de n-gramas sea el más utilizado

- Las probabilidades de sus eventos elementales se estiman con facilidad, éstos modelos básicamente realizan conteos de palabras.
- El modelo es fácil de implementar y su formulación es muy sencilla.
- Los algoritmos de estimación son muy eficientes.
- El modelo captura adecuadamente las relaciones entre palabras cercanas.

Desventajas:

- Por razones computacionales, el valor de n se debe mantener bajo ($n \leq 4$), de manera que solamente captura información inmediata; sin embargo, experimentos realizados han demostrado que cuando $n > 5$ el aporte no es considerable.
- Los n-gramas no se acomodan a los cambios dinámicos en el discurso, las frecuencias relativas de los n-gramas reflejan promedios sobre el corpus de entrenamiento, sin embargo cuando hay cambio de temática en el discurso, es posible que para algunas palabras la probabilidad de ocurrencia se vea modificada.
- El modelo tiene problemas de dispersión, hay una gran cantidad de n-gramas que no son vistos en los datos de entrenamiento y por tanto la frecuencia relativa de ellos es cero, lo cual implicaría que se le asigne probabilidad cero (0) a una frase que contenga uno de tales n-gramas.

2.3. Suavizado en los modelos de n-gramas

Como se mencionó en la sección 2.2, una de las desventajas del modelo de n-gramas es que se encuentra una alta cantidad de n-gramas con frecuencia cero; para resolver este tipo de problemas se han desarrollado técnicas denominadas técnicas de suavizado, las cuales se estudiarán en la presente sección. El uso del suavizado produce distribuciones más uniformes incrementando los valores de aquellas probablidades muy bajas y reduciendo los valores de las probabilidades muy altas, aumentando así la capacidad predictiva de los modelos.

En los datos de entrenamiento definidos en (1), al calcular la probabilidad p(Cher lee un libro) y usando un modelo de bigramas se tiene:

$$p(\text{Cher lee un libro}) = p(Cher \mid \langle \boldsymbol{BOS} \rangle) p(lee \mid Cher) \text{p}(un \mid lee) \\ p(libro \mid un) p(\langle \boldsymbol{eos} \rangle \mid libro).$$

$$p(Cher \mid \langle \boldsymbol{BOS} \rangle) = \frac{\mathbf{c}(\langle \mathbf{BOS} \rangle \ Cher)}{\sum_w \mathbf{c}(\langle \mathbf{BOS} \rangle w)} = \frac{0}{3}$$

$$p(lee \mid Cher) = \frac{\mathbf{c}(Cher \ lee)}{\sum_w \mathbf{c}(Cher \ w)} = \frac{\mathbf{0}}{1}$$

$$p(un \mid lee) = \frac{\mathbf{c}(lee \ un)}{\sum_w \mathbf{c}(lee \ w)} = \frac{2}{3}$$

$$p(libro \mid un) = \frac{\mathbf{c}(un \ libro)}{\sum_w \mathbf{c}(un \ w)} = \frac{1}{2}$$

$$p(\langle \textbf{\textit{eos}} \rangle \mid libro) = \frac{\mathbf{c}(libro \langle \textbf{eos} \rangle)}{\sum_w \mathbf{c}(libro\ w)} = \frac{1}{2}$$

con lo que:

$$\begin{aligned} p(\text{Cher lee un libro}) &= \frac{0}{3} \times 0 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \\ &= 0. \end{aligned}$$

Se debe tener en cuenta que en el sistema de reconocimiento de la voz, se desea encontrar la frase $\hat{w}$ en W^+ que tiene la mayor probabilidad de ocurrir dada la señal acústica O, es decir:

$$\hat{w} = Arg \left(\max_{w \in W^+} p(w \mid O) \right)$$

$$= Arg \left(\max_{w \in W^+} \frac{p(O \mid w) p(w)}{p(O)} \right).$$

Si $p(w)$ es cero, la cadena w no será considerada como una transcripción, sin tener en cuenta cuan clara es la señal acústica. Así, siempre que una frase w sea tal que $p(w) = 0$ durante una tarea de reconocimiento del habla, se producirá un error. Asignando a todas las frases una probabilidad no cero se previenen errores en el reconocimiento del habla. El suavizado es usado para solucionar este problema y describe técnicas para ajustar la estimación de probabilidades. El nombre Suavizado viene del hecho que estas técnicas tienden a hacer las distribuciones más uniformes, las técnicas de suavizado no solamente previenen probabilidades cero, si no que también mejoran la calidad del modelo.

La técnica de suavizado más simple consiste en sumar 1 (uno) a la frecuencia de aparición de cada bigrama en el corpus de datos (este método fue desarrollado por Lidstone, 1920; Johnson, 1932; Jeffreys, 1948) en [1], es decir:

$$\begin{aligned} p(w_i \mid w_{i-1}) &= \frac{1 + \mathbf{c}(w_{i-1}w_i)}{\sum_{w_i}[1 + \mathbf{c}(w_{i-1}w_i)]} \\ &= \frac{1 + \mathbf{c}(w_{i-1}w_i)}{|W| + \sum_{w_i}\mathbf{c}(w_{i-1}w_i)}, \end{aligned} \tag{2.7}$$

Aplicando la técnica anterior para estimar nuevamente p(Jhon lee un libro) y p(Cher lee un libro), con el conjunto de entrenamiento E formado por las frases dadas en (1), con W el vocabulario extraído de E, $|W| = 11$ que es el número de palabras diferentes que aparecen en E. Así:

$$\begin{aligned} p(\text{Jhon lee un libro}) &= p(Jhon \mid \langle \textbf{\textit{BOS}} \rangle)p(lee \mid Jhon)\text{p}(un \mid lee) \\ &\quad p(libro \mid un)p(\langle \textbf{\textit{eos}} \rangle \mid libro). \\ &= \frac{1+1}{11+3} \times \frac{1+1}{11+1} \times \frac{1+2}{11+3} \times \frac{1+1}{11+2} \times \frac{1+1}{11+2} \\ &= \frac{2}{14} \times \frac{2}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \\ &\approx 0{,}0001. \end{aligned}$$

En otras palabras, se estima que la frase Jhon lee un libro ocurre una vez cada 10 mil frases, esto es mucho más razonable que la estimación de la probabilidad de 0.06, es decir, que esta frase ocurre una vez cada 17 frases. Para la frase Cher lee un libro se tiene:

$$\begin{aligned} p(\text{Cher lee un libro}) &= p(Cher \mid \langle \textbf{\textit{BOS}} \rangle)p(lee \mid Cher)\text{p}(un \mid lee) \\ &\quad p(libro \mid un)p(\langle \textbf{\textit{eos}} \rangle \mid libro). \end{aligned}$$

$$= \frac{1+0}{11+3} \times \frac{1+0}{11+1} \times \frac{1+2}{11+3} \times \frac{1+1}{11+2} \times \frac{1+1}{11+2}$$

$$= \frac{1}{14} \times \frac{1}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13}$$

$$\approx 0{,}00003.$$

De nuevo, esto es más razonable que la probabilidad cero asignada por el modelo sin suavizar.

Existen varios algoritmos de suavizado para los modelos de n-gramas, a continuación se mostrarán los usados en la mayoría de los casos.

2.3.1. Suavizado Aditivo

Uno de los tipos de suavizado más simple usado en la práctica es el suavizado aditivo (desarrollado por Lidstone, 1920; Johnson, 1932; Jeffreys, 1948) mencionados en [1], el cual es simplemente una generalización del método dado en la ecuación 2.7. En lugar de pretender que cada n-grama ocurra una vez más de lo que ocurre, se pretende que éste ocurra δ veces más de lo que ocurre, donde $0 < \delta < 1$, es decir:

$$\begin{aligned} p_{add}(w_i \mid w_{i-1}) &= \frac{\delta + \mathbf{c}(w_{i-1}w_i)}{\sum_{w_i} [\delta + \mathbf{c}(w_{i-1}w_i)]} \\ &= \frac{\delta + \mathbf{c}(w_{i-1}w_i)}{\delta \, |W| + \sum_{w_i} \mathbf{c}(w_{i-1}w_i)} \end{aligned} \tag{2.8}$$

Cuando $\delta = 1$ se tiene el suavizado mostrado en (2.7). Gale y Church nombrados en [1] afirman que en este caso el método no es muy bueno.

2.3.2. Estimación Good-Turing

La aproximación Good-Turing en [1], es central en muchas técnicas de suavizado. Esta aproximación dice que para cualquier n-grama que ocurra r veces, se puede suponer que este ocurra r^* veces, siendo

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{2.9}$$

y donde n_r es el número de n-gramas que ocurren exactamente r veces en los datos de entrenamiento. Para convertir este conteo a una probabilidad se normaliza, luego para un n-grama α que ha ocurrido r veces se define la probabilidad de Good-Turing (p_{GT}) como

$$p_{GT}(\alpha) = \frac{r^*}{N} \tag{2.10}$$

donde

$$N = \sum_{r=0}^{\infty} n_r r^*.$$

Note que

$$N = \sum_{r=0}^{\infty} n_r r^* \;=\; \sum_{r=0}^{\infty} n_r (r+1)\frac{n_{r+1}}{n_r} \;=\; \sum_{r=0}^{\infty} (r+1) n_{r+1}$$

$$= \sum_{r=1}^{\infty} r n_r.$$

N es igual al número total de ocurrencias de los n-gramas en el corpus.

Veamos cómo se obtiene la estimación de Good-Turing dada en la ecuación (2.10).

Sea un total de s diferentes n-gramas α_1, α_2,..., α_s con frecuencias p_1, p_2,..., p_s respectivamente. Para estimar la verdadera probabilidad de un n-grama α_i que ocurre r veces en el corpus de datos, dado que no se conoce la identidad de este n-grama, pero se tienen las probabilidades p_1, p_2,..., p_s, ésta probabilidad se puede calcular pensando en el valor esperado de que p_i sea la probabilidad del n-grama α_i dado que este ocurre r veces, lo cual se puede expresar como la esperanza condicional de p_i dado que $c(\alpha_i) = r$ teniendo en cuenta que tanto los n-gramas α_1, α_2,..., α_s y las probabilidades p_1, p_2,..., p_s son variables aleatorias discretas se expresa de la siguiente manera:

$$\begin{aligned} E(p_i \mid \mathbf{c}(\alpha_i) = r) &= \sum_{j=1}^{s} p_j p(p_j \mid \mathbf{c}(\alpha_i) = r) \\ &= \sum_{j=1}^{s} p_j p(i = j \mid \mathbf{c}(\alpha_i) = r). \end{aligned} \tag{2.11}$$

La probabilidad $p(i = j \mid \mathbf{c}(\alpha_i) = r)$ es la probabilidad que un n-grama desconocido α_i que ocurre r veces es en verdad el j-ésimo n-grama α_j (con frecuencia correspondiente p_j). Esto se puede reescribir como:

$$\begin{aligned} p(i = j \mid \mathbf{c}(\alpha_i) = r) &= \frac{p(\mathbf{c}(\alpha_i) = r)}{\sum_{j=1}^{s} p(\mathbf{c}(\alpha_i) = r)} \\ &= \frac{\binom{N}{r} p_j^r (1 - p_j)^{N-r}}{\sum_{j=1}^{s} \binom{N}{r} p_j^r (1 - p_j)^{N-r}}. \end{aligned} \tag{2.12}$$

Esta última igualdad se tiene dado que la probabilidad $p(\mathbf{c}(\alpha_i) = r)$ se puede expresar usando la fórmula de la probabilidad binomial:

$$p(r) = \binom{N}{r} p^r q^{N-r} \tag{2.13}$$

donde:
$N = \sum_{j=1}^{s} \mathbf{c}(\alpha_j)$ es el número total de ocurrencias y $p(r) = p(\mathbf{c}(\alpha_i) = r)$.

Simplificando en (2.12) se tiene:

$$p(i = j \mid \mathbf{c}(\alpha_i) = r) = \frac{\binom{N}{r} p_j^r (1 - p_j)^{N-r}}{\binom{N}{r} \sum_{j=1}^{s} p_j^r (1 - p_j)^{N-r}}. \tag{2.14}$$

Sustituyendo (2.13) en la ecuación (2.11) se tiene

$$\begin{aligned} E(p_i \mid \mathbf{c}(\alpha_i) = r) &= \sum_{j=1}^{s} p_j \left[p(i = j \mid \mathbf{c}(\alpha_i) = r) \right] \\ &= \sum_{j=1}^{s} p_j \left[\frac{\binom{N}{r} p_j^r (1 - p_j)^{N-r}}{\binom{N}{r} \sum_{j=1}^{s} p_j^r (1 - p_j)^{N-r}} \right] \\ &= \frac{\binom{N}{r} \sum_{j=1}^{s} p_j \left[p_j^r (1 - p_j)^{N-r} \right]}{\binom{N}{r} \sum_{j=1}^{s} \left[\sum_{j=1}^{s} p_j^r (1 - p_j)^{N-r} \right]} \\ &= \frac{\sum_{j=1}^{s} \binom{N}{r} p_j^{r+1} (1 - p_j)^{N-r}}{\sum_{j=1}^{s} \binom{N}{r} p_j^r (1 - p_j)^{N-r}}. \end{aligned} \tag{2.15}$$

Ahora, considerando $E_N(n_r)$ como el número esperado de n-gramas que ocurren r veces dado que el conteo total es N, esto es igual a la suma de la probabilidad de cada n-grama que aparece exactamente r veces, es decir:

$$\begin{aligned} E_N(n_r) &= \sum_{j=1}^{s} p(\mathbf{c}(\alpha_j) = r) \\ &= \sum_{j=1}^{s} \binom{N}{r} p_j^r (1 - p_j)^{N-r}. \end{aligned} \tag{2.16}$$

De igual manera

$$E_{N+1}(n_{r+1}) = \sum_{j=1}^{s} p(\mathbf{c}(\alpha_j) = r+1)$$

$$= \sum_{j=1}^{s} \binom{N+1}{r+1} p_j^{r+1}(1-p_j)^{N-r}$$

$$= \sum_{j=1}^{s} \frac{N!\,(N+1)}{r!\,(r+1)\,(N-r)!} p_j^{r+1}(1-p_j)^{N-r}$$

$$= \left[\frac{N+1}{r+1}\right] \sum_{j=1}^{s} \binom{N}{r} p_j^{r+1}(1-p_j)^{N-r}.$$

De la expresión anterior

$$\left[\frac{r+1}{N+1}\right] E_{N+1}(n_{r+1}) = \sum_{j=1}^{s} \binom{N}{r} p_j^{r+1}(1-p_j)^{N-r}. \tag{2.17}$$

En la expresión (2.15)

$$E(p_i \mid \boldsymbol{c}(\alpha_i) = r) = \frac{\sum_{j=1}^{s} \binom{N}{r} p_j^{r+1}(1-p_j)^{N-r}}{\sum_{j=1}^{s} \binom{N}{r} p_j^{r}(1-p_j)^{N-r}}$$

sustituyendo (2.16) y (2.17)

$$E(p_i \mid \boldsymbol{c}(\alpha_i) = r) = \left[\frac{r+1}{N+1}\right] \frac{E_{N+1}(n_{r+1})}{E_N(n_r)}.$$

La cual es una estimación aproximada para la probabilidad esperada de un n-grama α_i que se ha contado r veces. Para expresar lo anterior en términos de una estimación r^* se usa la ecuación (2.10) como sigue:

$$\frac{r^*}{N} = p(\alpha_i)$$

$$r^* = Np(\alpha_i)$$

$$= N\left[\frac{r+1}{N+1}\frac{E_{N+1}(n_{r+1})}{E_N(n_r)}\right]$$

$$\approx (r+1)\frac{n_{r+1}}{n_r}.$$

Nótese que en la ecuación anterior son usadas las aproximaciones $E_{N+1}(n_{r+1}) \approx n_{r+1}$, $E_N(n_r) \approx n_r$ y $N \approx N+1$. En otras palabras, usamos valores empíricos de n_r para estimar el valor esperado real.

La estimación de Good-Turing no puede usarse cuando $n_r = 0$, es decir cuando no hay n-gramas que ocurren r veces, en este caso es necesario suavizar n_r de tal forma que exista por lo menos un n-grama que ocurra r veces. En la práctica la estimación de Good-Turing no es usada por si sola para el suavizado de n-gramas, sin embargo se usa como una herramienta en varias técnicas de suavizado.

2.3.3. Suavizado de Jelineck-Mercer

Considere el caso de construir un modelo de bigramas con datos de entrenamiento donde se tiene que:

c(RECORRIÓ EL)= 0

c(RECORRIÓ SU)= 0

Entonces, según el suavizado aditivo se tiene que
$p(EL \mid RECORRIO) = p(SU \mid RECORRIO)$ puesto que

$$p_{add}(EL \mid RECORRIO) = \frac{\delta + \mathbf{c}(RECORRIO\, EL)}{\delta\,|W| + \sum_{w_i} \mathbf{c}(RECORRIO\, w_i)}$$

$$= \frac{\delta + 0}{\delta\,|W| + \sum_{w_i} \mathbf{c}(RECORRIO\, w_i)}$$

$$= \frac{\delta}{\delta\,|W| + \sum_{w_i} \mathbf{c}(RECORRIO\, w_i)}.$$

Por otro lado:

$$p_{add}(SU \mid RECORRIO) = \frac{\delta + \mathbf{c}(RECORRIO\, SU)}{\delta\,|W| + \sum_{w_i} \mathbf{c}(RECORRIO\, w_i)}$$

$$= \frac{\delta + 0}{\delta\,|W| + \sum_{w_i} \mathbf{c}(RECORRIO\, w_i)}$$

$$= \frac{\delta}{\delta\,|W| + \sum_{w_i} \mathbf{c}(RECORRIO\, w_i)}$$

Esto es algo que no refleja la realidad ya que la palabra *EL* es mucho más común en nuestro vocabulario que la palabra *SU*, esto implicaría que $p(EL \mid RECORRIO) > p(SU \mid RECORRIO)$. Para poder captar este comportamiento se puede realizar una interpolación (combinación) del modelo de

bigrama con el modelo de unigrama. Un modelo de unigrama condiciona la probabilidad a una palabra y sólo refleja la frecuencia de cada palabra en el texto, de ahí que la probabilidad de un modelo de unigrama es estimada por:

$$p_{ML}(w_i) = \frac{\mathbf{c}(w_i)}{\sum_{w_i} \mathbf{c}(w_i)}.$$

Realizando la interpolación de un modelo de bigrama con un modelo de unigrama se tiene:

$$p_{interp}(w_i \mid w_{i-1}) = \lambda p_{ML}(w_i \mid w_{i-1}) + (1-\lambda) p_{ML}(w_i), \tag{2.18}$$

donde $0 \leq \lambda \leq 1$ y $p_{ML}(w_i \mid w_{i-1})$ es el modelo en (2.4). Aplicando la interpolación al ejemplo anterior se tiene que:

$$p_{ML}(EL \mid RECORRIO) = \frac{\mathbf{c}(RECORRIO\ EL)}{\sum_{w_i} \mathbf{c}(RECORRIO\ w_i)}$$

$$= 0,$$

y

$$p_{ML}(SU \mid RECORRIO) = \frac{\mathbf{c}(RECORRIO\ SU)}{\sum_{w_i} \mathbf{c}(RECORRIO\ w_i)}$$

$$= 0.$$

Dado que $p_{ML}(EL) \gg p_{ML}(SU)$, se tiene que
$p_{interp}(EL \mid RECORRIO) > p_{interp}(SU \mid RECORRIO)$ puesto que por la

ecuación (2.16)

$$p_{interp}(EL \mid RECORRIO) = (1 - \lambda)p_{ML}(EL)$$

y

$$p_{interp}(SU \mid RECORRIO) = (1 - \lambda)p_{ML}(SU).$$

En general, se pueden interpolar modelos de n-gramas de orden superior con modelos de n-gramas de menor orden, porque donde hay insuficientes datos para estimar una probabilidad en el caso de orden superior, la de menor orden a menudo puede ofrecer información útil. Una elegante manera de realizar esta interpolación es la descrita por Brown [2] como sigue:

$$p_{interp}(w_i \mid w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} p_{ML}(w_i \mid w_{i-n+1}^{i-1}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{interp}(w_i \mid w_{i-n+2}^{i-1}),$$

luego, un modelo de interpolación de orden n se define como una interpolación lineal recursiva entre el modelo de probabilidad máxima (p_{ML}) de órden n y el modelo de interpolación de órden n-1.
Alternativamente Jelineck y Mercer describen una técnica conocida como Interpolación eliminada o Estimación eliminada donde partes diferentes de los datos de entrenamiento alternan en cualquiera de los dos, la p_{ML} o $\lambda_{w_{i-n+1}^{i-1}}$ y luego los resultados son promediados. Note que $\lambda_{w_{i-n+1}^{i-1}}$ será diferente para historias diferentes w_{i-n+1}^{i-1}.

2.3.4. Suavizado de Katz

El suavizado de Katz en [1] extiende los conceptos de Good-Turing agregando la combinación de modelos de orden superior con modelos de órden

más bajo. Para un modelo de bigrama w_{i-1}^{i} con $\mathbf{c}(w_{i-1}^{i}) = r$, el suavizado de Katz se calcula usando la ecuación:

$$C_{Katz}(w_{i-1}^{i}) = \begin{cases} d_r r, & \text{si } r > 0 \\ \alpha(w_{i-1}) p_{ML}(w_i), & \text{si } r = 0 \end{cases} \tag{2}$$

Es decir, todos los bigramas con un conteo r diferente de cero son disminuidos de acuerdo a un coeficiente de descuento d_r. El coeficiente de descuento d_r es aproximadamente $\frac{r^*}{r}$, la cual es la disminución prevista por la estimación de Good-Turing dada en la ecuación (2.9). El valor $\alpha(w_{i-1})$ es escogido de forma tal que el número total de conteos en la distribución $\sum_{w_i} C_{Katz}(w_{i-1}^{i})$ permanece inalterado, es decir,

$\sum_{w_i} C_{Katz}(w_{i-1}^{i}) = \sum_{w_i} C(w_{i-1}^{i})$, el valor apropiado para $\alpha(w_{i-1})$ es

$$\alpha(w_{i-1}) = \frac{1 - \sum_{w_i : C(w_{i-1}^{i})>0} p_{Katz}(w_i \mid w_{i-1})}{\sum_{w_i : C(w_{i-1}^{i})=0} p_{ML}(w_i)}$$

$$= \frac{1 - \sum_{w_i : C(w_{i-1}^{i})>0} p_{Katz}(w_i \mid w_{i-1})}{1 - \sum_{w_i : C(w_{i-1}^{i})>0} p_{ML}(w_i)}.$$

Para calcular $p_{Katz}(w_i \mid w_{i-1})$ del conteo de ésta última igualdad sólo se normaliza

$$p_{Katz}(w_i \mid wi - 1) = \frac{C_{Katz}(w_{i-1}^{i})}{\sum_{w_i} C_{Katz}(w_{i-1}^{i})}.$$

Los d_r son calculados de la siguiente manera, los conteos para un valor de r grande son tomados ya que no se descuentan, en particular Katz toma $d_r = 1$ para todo $r > 5$. Los coeficientes de disminución para conteos más bajos ($r \leq 5$) son obtenidos de la estimación de Good-Turing aplicada a la distribución global de bigramas; es decir, el n_r en la ecuación (2.9) indica el número total de bigramas que ocurren exactamente r veces en los datos de entrenamiento. Estos d_r son elegidos de forma que los descuentos resultantes son proporcionales a las distribuciones pronosticadas por la estimación de Good-Turing, cuando $r > 5$, la ecuación correspondiente es:

$$1 - d_r = \mu \left(1 - \frac{r^*}{r}\right), \tag{2.19}$$

para alguna constante $\mu > 0$.

Para el caso $r \leq 5$, los d_r son elegidos de forma tal que el número total de conteos disminuidos en la distribución global del bigrama es igual al número total de conteos que deben ser asignados a los bigramas con conteos cero de acuerdo con la estimación de Good-Turing [3].

La estimación de Good-Turing predice que el número total de conteos que deben ser asignados a bigramas con conteos cero es $n_0 0^* = n_1$ ya que al

[3]En la estimación de Good-Turing, el número total de conteos que se han hecho de los n-gramas con conteos diferentes a cero puede ser igual al número total de conteos asignados a n-gramas con conteos cero. por lo tanto la constante normalización para una distribución suavizada es idéntica a la distribución original. El suavizado de Katz trata de conseguir un efecto similar excepto solamente en la reducción completa de conteos donde $r \leq 5$

reemplazar $r = 0$ en la siguiente ecuación se tiene:

$$r^* = (r+1)\frac{n_{r+1}}{n_r}$$

$$0^* = (0+1)\frac{n_{0+1}}{n_0}$$

$$0^* = \frac{n_1}{n_0}$$

$$n_0 0^* = n_1.$$

Así que para el caso $r \leq 5$ la ecuación correspondiente es:

$$\sum_{r=1}^{k}(1-d_r)r = n_1. \tag{2.20}$$

La solución para las ecuaciones (2.19) y (2.20) está dada por

$$d_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}}.$$

Como se muestra en la ecuación (2), el modelo de bigrama es definido en términos del modelo de unigrama; en general, el modelo de suavizado de Katz de orden n es definido en términos del modelo de suavizado de Katz de orden $n-1$, de manera similar al suavizado de Jelineck-Mercer. Para terminar la recursión, el modelo de unigrama de Katz es tomado como el modelo de máxima verosimilitud del modelo de unigrama.

2.3.5. Suavizado de Witten-Bell

El suavizado de Witten-Bell fue desarrollado para trabajar en la comprensión de textos y puede ser considerado como un ejemplo del suavizado de Jelineck-Mercer. En particular, el modelo de suavizado de orden n-ésimo es definido recursivamente como una interpolación lineal entre el modelo de probabilidad máxima de orden n y el modelo del suavizado de Witten-Bell (WB) de orden n-1 como sigue:

$$\begin{aligned} p_{WB}(w_i \mid w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} p_{ML}(w_i \mid w_{i-n+1}^{i-1}) \\ + (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{WB}(w_i \mid w_{i-n+2}^{i-1}). \end{aligned} \tag{2.21}$$

Para computar los parámetros $\lambda_{w_{i-n+1}^{i-1}}$ para el suavizado de Witten-Bell se usa un número de palabras únicas que siguen a la historia w_{i-n+1}^{i-1}, este valor se notará como $N_{1+}w_{i-n+1}^{i-1}\bullet$ definido así:

$$N_{1+}w_{i-n+1}^{i-1}\bullet = \left|\left\{ w_j : C\left(w_{i-n+1}^{i-1} w_j\right) > 0 \right\}\right|. \tag{2.22}$$

La notación N_{1+} es representada para indicar el número de palabras que tienen uno o más conteos y el $\bullet$ indica una variable libre sobre la que se realiza la suma. La asignación a los parámetros $1 - \lambda_{w_{i-n+1}^{i-1}}$ en el suavizado de Witten-Bell es tal que:

$$1 - \lambda_{w_{i-n+1}^{i-1}} = \frac{N_{1+}w_{i-n+1}^{i-1}\bullet}{N_{1+}w_{i-n+1}^{i-1}\bullet + \sum_{w_i} C(w_{i-n+1}^{i})}, \tag{2.23}$$

sustituyendo la ecuación (2.23) en la ecuación (2.21) se tiene:

$$\begin{aligned} p_{WB}(w_i \mid w_{i-n+1}^{i-1}) = 1 - \frac{N_{1+}w_{i-n+1}^{i-1}\bullet}{N_{1+}w_{i-n+1}^{i-1}\bullet + \sum_{w_i} C(w_{i-n+1}^{i})} p_{ML}(w_i \mid w_{i-n+1}^{i-1}) \\ + \frac{N_{1+}w_{i-n+1}^{i-1}\bullet}{N_{1+}w_{i-n+1}^{i-1}\bullet + \sum_{w_i} C(w_{i-n+1}^{i})} p_{WB}(w_i \mid w_{i-n+2}^{i-1}) \end{aligned}$$

Se nota $p_{ML}(w_i \mid w_{i-n+1}^{i-1})$ como A, luego

$$p_{WB}(w_i \mid w_{i-n+1}^{i-1}) = \frac{\left[N_{1+}w_{i-n+1}^{i-1}\bullet + \sum_{w_i} C(w_{i-n+1}^{i}) - N_{1+}w_{i-n+1}^{i-1}\bullet\right]}{N_{1+}w_{i-n+1}^{i-1}\bullet + \sum_{w_i} C(w_{i-n+1}^{i})}\ A$$

$$+ \frac{N_{1+}w_{i-n+1}^{i-1}\bullet\ p_{WB}(w_i \mid w_{i-n+2}^{i-1})}{N_{1+}w_{i-n+1}^{i-1}\bullet + \sum_{w_i} C(w_{i-n+1}^{i})}$$

$$= \frac{\sum_{w_i} C(w_{i-n+1}^{i})\ A + N_{1+}w_{i-n+1}^{i-1}\bullet\ p_{WB}(w_i \mid w_{i-n+2}^{i-1})}{N_{1+}w_{i-n+1}^{i-1}\bullet + \sum_{w_i} C(w_{i-n+1}^{i})}$$

Dado que $p_{ML}(w_i \mid w_{i-n+1}^{i-1}) = \dfrac{C(w_{i-n+1}^{i})}{\sum_{w_i} C(w_{i-n+1}^{i})}$ se tiene:

$$p_{WB}(w_i \mid w_{i-n+1}^{i-1}) = \frac{C(w_{i-n+1}^{i}) + N_{1+}w_{i-n+1}^{i-1}\bullet\ p_{WB}(w_i \mid w_{i-n+2}^{i-1})}{N_{1+}w_{i-n+1}^{i-1}\bullet + \sum_{w_i} C(w_{i-n+1}^{i})} \qquad (2.24)$$

2.3.6. Reducción total

La reducción absoluta o total, como en el suavizado de Jelineck-Mercer, involucra la interpolación de modelos de más alto y bajo orden. Sin embargo, en vez de multiplicar la distribución del más alto orden de la probabilidad máxima por el factor $\lambda_{w_{i-n+1}^{i-1}}$, la distribución de más alto orden es creada por una resta de disminución fija $D \leq 1$ de cada conteo diferente de cero y en lugar de la ecuación:

$$p_{interp}(w_i \mid w_{i-n+1}^{i-1}) = \lambda_{w_{i-n+1}^{i-1}} p_{ML}(w_i \mid w_{i-n+1}^{i-1}) + (1 - \lambda_{w_{i-n+1}^{i-1}}) p_{interp}(w_i \mid w_{i-n+2}^{i-1})$$

se tiene la siguiente expresión:

$$p_{abs}(w_i \mid w_{i-n+1}^{i-1}) = \frac{\text{máx}\left\{C(w_{i-n+1}^{i}) - D, 0\right\}}{\sum_{w_i} C(w_{i-n+1}^{i}) + (1 - \lambda_{w_{i-n+1}^{i-1}})} p_{abs}(w_i \mid w_{i-n+2}^{i-1}). \quad (2.25)$$

Ney, Essen y Kneser en [1] sugirieron estimar el valor D de la siguiente manera:

$$D = \frac{n_1}{n_1 + 2n_2}$$

donde n_1 y n_2 son el número total de n-gramas con exactamente uno y dos conteos respectivamente.

2.3.7. Suavizado de Kneser-Ney

Considerando el desarrollo de un modelo de bigrama en los datos donde existe una palabra muy común, por ejemplo Francisco que se produce sólo despues a una sola palabra, San, dado que C(Francisco) es alto, la probabilidad del unigrama $p(Francisco)$ será alta y la reducción absoluta asignará una probabilidad alta a la palabra Francisco. Sin embargo esta probabilidad no debe ser alta ya que en los datos de entrenamiento la palabra Francisco tiene solamente una historia. Es decir, quizá la palabra Francisco debe recibir una probabilidad baja de unigrama porque la única vez en que la palabra aparece es cuando la última palabra es San, en tal caso la probabilidad del modelo de bigrama es buena probabilidad. Asignando un conteo al unigrama correspondiente siempre que tal evento ocurra, entonces el número de conteos asignados a cada unigrama será solamente el número de palabras diferentes que le siguen. En el suavizado de Kneser-Ney (p_{KN}) la probabilidad de unigrama en un modelo de bigrama es calculado de esta manera, Por ejemplo, para un modelo de bigrama, una distribución de suavizado p_{KN} sobre el unigrama anterior será:

$$\sum_{w_{i-1}} p_{KN}(w_{i-1}w_i) = \frac{C(w_i)}{\sum_{w_i} C(w_i)} \quad (2.26)$$

para todo w_i.

Al lado izquierdo de la ecuación anterior está el suavizado p_{KN} de la distribución del bigrama $w_{i-1}w_i$ sobre todos los unigramas anteriores a w_i y al lado derecho está la frecuencia del unigrama w_i encontrada en los datos de entrenamiento. La ecuación siguiente muestra la definición del suavizado de Kneser-Ney.

$$p_{KN}(w_i \mid w_{i-n+1}^{i-1}) = \frac{\text{máx}\left\{C(w_{i-n+1}^{i}) - D, 0\right\}}{\sum_{w_i} C(w_{i-n+1}^{i})} + \frac{D}{\sum_{w_i} C(w_{i-n+1}^{i})} N_{1+}w_{i-n+1}^{i-1} \bullet p_{KN}(w_i \mid w_{i-n+2}^{i-1}). \tag{2.27}$$

Otra forma de ver el suavizado es:

$$p_{KN}(w_i \mid w_{i-1+1}^{i-1}) = \begin{cases} \dfrac{\text{máx}\left\{C(w_{i-n+1}^{i}) - D, 0\right\}}{\sum_{w_i} C(w_{i-n+1}^{i})}, & \text{si } C(w_{i-n+1}^{i}) > 0 \\ \gamma(w_{i-n+1}^{i-1}) p_{KN}(w_{i-n+2}^{i-1}), & \text{si } C(w_{i-n+1}^{i}) = 0 \end{cases}$$

donde $\gamma(w_{i-n+1}^{i-1})$ es elegido para hacer que la distribución sume 1. Es decir se hace una interpolación en la distribución de más bajo orden no sólo con las palabras que tienen conteo cero en la distribución de más alto órden si no con todas las palabras existentes en el corpus.

Expandiendo la ecuación (2.26) se tiene

$$\frac{C(w_i)}{\sum_{w_i} C(w_i)} = \sum_{w_{i-1}} p_{KN}(w_i \mid w_{i-1}) p(w_{i-1}) \tag{2.28}$$

para $p(w_{i-1})$ tomando la distrubución encontrada en los datos de entrenamiento se tiene

$$p(w_{i-1}) = \frac{C(w_{i-1})}{\sum_{w_{i-1}} C(w_{i-1})}$$

reemplazando la ecuación anterior en la ecuación (2.27) se tiene

$$\frac{C(w_i)}{\sum_{w_i} C(w_i)} = \sum_{w_{i-1}} p_{KN}(w_i \mid w_{i-1}) \frac{C(w_{i-1})}{\sum_{w_{i-1}} C(w_{i-1})}$$

$$\sum_{w_{i-1}} C(w_{i-1}) \frac{C(w_i)}{\sum_{w_i} C(w_i)} = \sum_{w_{i-1}} p_{KN}(w_i \mid w_{i-1}) C(w_{i-1})$$

$$\frac{C(w_i)}{\sum_{w_{i-1}} C(w_{i-1})} = \sum_{w_{i-1}} p_{KN}(w_i \mid w_{i-1}) C(w_{i-1})$$

$$C(w_i) = \sum_{w_{i-1}} C(w_{i-1}) \sum_{w_{i-1}} p_{KN}(w_i \mid w_{i-1}) C(w_{i-1})$$

$$C(w_i) = \sum_{w_{i-1}} C(w_{i-1}) p_{KN}(w_i \mid w_{i-1})$$

sustituyendo el valor de p_{KN} de la ecuación (2.27) en la ecuación anterior

$$\begin{aligned} C(w_i) &= \sum_{w_{i-1}} C(w_{i-1}) p_{KN}(w_i \mid w_{i-1}) \\ &= \sum_{w_{i-1}} C(w_{i-1}) \left[\frac{\text{máx}\left\{C(w_{i-1}w_i) - D, 0\right\}}{\sum_{w_i} C(w_{i-1}w_i)} \right] + \\ &\quad \sum_{w_{i-1}} C(w_{i-1}) \left[\frac{D}{\sum_{w_i} C(w_{i-n+1}^{i})} \quad N_{1+}(w_{i-1}\bullet) p_{KN}(w_i) \right] \end{aligned} \tag{2.29}$$

La expresión máx $\{C(w_{i-1}w_i) - D, 0\}$ significa que se toman las frases que han ocurrido cero, una o más veces en los datos de entrenamiento, dado el

caso en el que máx $\{C(w_{i-1}w_i) - D, 0\} > 0$ la expresión se puede reescribir como la suma sobre las historias de la palabra w_i talque $w_{i-1}w_i$ ha ocurrido al menos una vez, es decir $\sum_{w_{i-1}:C(w_{i-1}w_i)>0}$, así la ecuación (2.26) queda

$$
\begin{aligned}
C(w_i) = & \sum_{w_{i-1}:C(w_{i-1}w_i)>0} C(w_{i-1})\frac{C(w_{i-1}w_i) - D}{C(w_{i-1})} + \\
& \sum_{w_{i-1}} C(w_{i-1})\frac{D}{C(w_{i-1})}N_{1+}(w_{i-1}\bullet)p_{KN}(w_i) \\
= & \sum_{w_{i-1}:C(w_{i-1}w_i)>0} (C(w_{i-1}w_i) - D) + \sum_{w_{i-1}} DN_{1+}(w_{i-1}\bullet)p_{KN}(w_i) \\
= & \sum_{w_{i-1}:C(w_{i-1}w_i)>0} C(w_{i-1}w_i) \; - \sum_{w_{i-1}:C(w_{i-1}w_i)>0} D + \\
& \sum_{w_{i-1}} DN_{1+}(w_{i-1}\bullet)p_{KN}(w_i) \\
= & \, C(w_i) - \sum_{w_{i-1}:C(w_{i-1}w_i)>0} D + Dp_{KN}(w_i)\sum_{w_{i-1}} N_{1+}(w_{i-1}\bullet) \\
= & \, C(w_i) - N_{1+}(\bullet w_i)D + Dp_{KN}(w_i)N_{1+}(\bullet\bullet),
\end{aligned}
\tag{2.30}
$$

donde:

$$
\begin{aligned}
N_{1+}(\bullet w_i) = & \sum_{w_{i-1}:C(w_{i-1}w_i)>0} \\
= & \, |w_{i-1} : C(w_{i-1}w_i) > 0|
\end{aligned}
$$

es el número de palabras diferentes w_{i-1} que precede a w_i en los datos de entrenamiento y la expresión $N_{1+}(\bullet\bullet)$ es:

$$N_{1+}(\bullet\bullet) = \sum_{w_{i-1}} N_{1+}(w_{i-1}\bullet)$$

$$= |(w_{i-1}, w_i) : C(w_{i-1}w_i) > 0|$$

$$= \sum_{w_i} N_{1+}(\bullet w_i).$$

Despejando $p_{KN}(w_i)$ de la ecuación (2.30)

$$C(w_i) = C(w_i) - N_{1+}(\bullet w_i)D + Dp_{KN}(w_i)N_{1+}(\bullet\bullet)$$

$$C(w_i) - C(w_i) + N_{1+}(\bullet w_i)D = Dp_{KN}(w_i)N_{1+}(\bullet\bullet)$$

$$N_{1+}(\bullet w_i) = p_{KN}(w_i)N_{1+}(\bullet\bullet)$$

$$\frac{N_{1+}(\bullet w_i)}{N_{1+}(\bullet\bullet)} = p_{KN}(w_i).$$

Capítulo 3

Modelos Mixtos

Como se mencionó en el capítulo anterior, el modelo de lenguaje que ha demostrado mayor eficiencia y sencillez en su implementación es el de n-gramas. También se observaron algunas de sus limitaciones, entre otras, que sólo utiliza información de pocas palabras para calcular la probabilidad, es decir, información a corta distancia. Una propuesta para contrarrestar esta limitación es combinar el modelo mediante algún método con otros modelos, con lo cual el modelo resultante de esta combinación es un *MODELO HIBRIDO o MIXTO*.
En este capítulo se va a estudiar la combinación del modelo de n-gramas con modelos de lenguaje como son los basados en gramáticas y los basados en el principio de máxima entropía.

3.1. Introducción

Las gramáticas formales son una herramienta potente para la generación de lenguajes basados en características sintácticas. Las gramáticas incontextuales probabilísticas (GI) constituyen un tipo importante de modelos ya que permiten representar dependencias a larga distancia entre los elementos. Como se mencionó en el capítulo 2, dentro de los inconvenientes que presenta

el modelo de n-gramas es que éste considera información local, básicamente relaciones estadísticas entre algunas pocas palabras descartando posible información contenida en la frase. Las GI por su parte son capaces de representar de una forma compacta las relaciones sintácticas que a largo plazo se establecen entre las palabras del lenguaje.
Una gramática incontextual probabilística (GIP) es una extensión natural de una GI que se compone básicamente de dos partes: un conjunto de reglas (gramática característica) que conforman la parte estructural de la misma y unas funciones de distribución de probabilidad asociadas a las reglas.
Se hará una combinación del modelo de n-gramas y el generado por las GI con el objetivo de combinar la característica principal de las GI que es la de aportar información a larga distancia con la característica fuerte de los n-gramas que aporta información a corta distancia. En la sección 3.2.1 se explica la forma de realizar ésta combinación.

Por otro lado el concepto de entropía, desde el punto de vista de la teoría de la información, es la medida de la cantidad de información que se puede codificar en una variable aleatoria X; a cada posible valor de esta variable se le asigna una cierta combinación de dígitos binarios, como fue detallado en el capítulo 1. En la sección 3.2.2 se detallará la manera en que se combina la entropía con la n-grama.

3.2. Combinación de n-gramas y GIP para el modelo de lenguaje

Una gramática incontextual GI es una cuádrupla (Σ, P, NT, S), donde Σ es un conjunto finito no vacío que representa el vocabulario, P un conjunto de reglas definadas de acuerdo con el campo de aplicación, NT un conjunto no vacío, finito, denominado de no terminales y S es el símbolo de inicio que está presente en toda gramática.

Una gramática incontextual probabilística (GIP) G_p, es una pareja (G, P) tal que G es una gramática incontextual y p una función $p : P \longrightarrow (0, 1]$ sobre las reglas de la gramática de tal forma que:

$$\forall A \in NT, \quad \sum_{(A \to \alpha)} p\,(A \to \alpha) = 1$$

Para mayor claridad, veamos un ejemplo en el que una gramática genera una frase. Sea: (Σ, ρ, NT, S) la gramática tal que:

$\Sigma = \{a, b\}$, $\quad NT = \{A, B, C, S\}$ y P el conjunto de reglas definidas así:

1. $S \to AB$
2. $S \to BC$
3. $A \to BA$
4. $A \to a$
5. $B \to BC$
6. $B \to b$
7. $C \to AB$
8. $C \to a$

donde la probabilidad de cada regla de la gramática se enuncia a continuación y cuya asignación según sea la aplicación se decidirá si es arbitraria o no, para este ejemplo será arbitraria.

1. $p(s \to AB) \;=\; 0{,}5$
2. $p(s \to BC) \;=\; 0{,}5$
3. $p(A \to BA) \;=\; 0{,}6$
4. $p(A \to a) \;=\; 0{,}4$
5. $p(B \to BC) \;=\; 0{,}7$

6. $p(B \rightarrow b) \ = \ 0{,}3$

7. $p(C \rightarrow AB) \ = \ 0{,}5$

8. $p(C \rightarrow a) \ = \ 0{,}5$

donde $S \Rightarrow BC$ significa que el no terminal S deriva en la secuencia de no terminales BC, además es de notar que los símbolos a y b son terminales ya que estos no derivan en otro símbolo.

Veamos una posible frase generada por la GIP. Partiendo del símbolo inicial S se tienen dos opciones, o derivar en AB o derivar en BC, elijamos la primera opción S deriva en AB, notado $S \Rightarrow AB$, en seguida se deriva el símbolo no terminal más a la izquierda en el consecuente, es decir se deriva A, nuevamente éste símbolo cuenta con dos posibles derivaciones y tendremos en cuenta para nuestro ejemplo su derivación en BA, así el resultado será $AB \Rightarrow BAB$; derivamos el símbolo no terminal más a la izquierda, es decir B, utilizando la regla 6 se tiene $BAB \Rightarrow bAB$; continuando con el mismo proceso usando las reglas 1, 3, 6, 4, 5, 6, 7, 4 y 6 la derivación final será:

$$S \rightarrow AB \rightarrow BAB \rightarrow bAB \rightarrow baB \rightarrow baBC \rightarrow babAB \rightarrow babab$$

así, mediante el uso de las reglas definidas en la gramática se ha generado la frase *babab*. Otra derivación para la frase *babab* se da a continuación:

$$S \rightarrow BC \rightarrow BCC \rightarrow bCC \rightarrow bABC \rightarrow baBC \rightarrow babC \rightarrow babAB \rightarrow babaB \rightarrow babab$$

$$p(x \mid G_p) = \sum_{d_x \in D_x} \prod_{(A \rightarrow \alpha) \in \rho} p(A \longrightarrow \alpha)^{N(A \longrightarrow \alpha, d_x)}$$

donde x es la frase generada por la GIP, D_x el conjunto formado por todas las derivaciones a izquierda de la cadena x y $N\,(A \rightarrow \alpha)$ es el número de

veces que la regla $A \rightarrow \alpha$ se ha utilizado en la derivación d_x.

Para obtener la probabilidad de *babab* se suman las probabilidades obtenidas en cada una de las 2 posibles derivaciones así:

$$p(x \mid G) = \prod p(A \rightarrow \alpha)^{N(A \rightarrow \alpha)} + \prod p(A \rightarrow \beta)^{N(A \rightarrow \beta)}$$

Luego la probabilidad de la frase $x = babab$ según la primera derivación será:

$$p(x; d_{1x} \mid G_p) = 0{,}5 \times 0{,}6 \times 0{,}3 \times 0{,}4 \times 0{,}7 \times 0{,}3 \times 0{,}5 \times 0{,}4 \times 0{,}3 \quad = 0{,}0005$$

y para la segunda derivación:

$$p(x; d_{2x} \mid G_p) = 0{,}5 \times 0{,}7 \times 0{,}3 \times 0{,}5 \times 0{,}4 \times 0{,}3 \times 0{,}5 \times 0{,}4 \times 0{,}3 \quad = 0{,}0004$$

Luego de la ecuación anterior se tiene:

$$p(x \mid G_p) = p(x; d_{1x} \mid G_p) + p(x; d_{2x} \mid G_p) \quad = 0{,}0005 + 0{,}0004 = 0{,}0009$$

Para poder combinar el modelo de n-gramas con un modelo generado por una gramática es necesario el cálculo de la probabilidad de la frase w generada por la gramática, es decir $p(w)$.
Un problema importante asociado con el modelado del lenguaje consiste en evaluar la siguiente expresión:

$$p(w_i \mid w_1 w_2 \ldots w_{i-1}). \tag{3.1}$$

Para calcular la expresión (3.1) es habitual simplificarla imponiendo restricciones sobre la historia de w_i. La restricción más común para predecir la palabra w_i es limitar la historia anterior a un pequeño conjunto de palabras. Los modelos de n-gramas permiten aproximar la expresión (3.1) calculando la probabilidad de la siguiente palabra a observar, considerando únicamente las $n - 1$ palabras anteriores:

$$p(w_i \mid w_{i-n+1} \dots w_{i-1}). \tag{3.2}$$

Otra forma de aproximar la expresión (3.1) consiste en utilizar una GIP. Las GIP son una alternativa adecuada para representar relaciones sintácticas sobre las palabras. Con este tipo de modelos, la expresión (3.1) toma la forma:

$$p(w_i \mid w_1 w_2 \dots w_{i-1}, G_p), \tag{3.3}$$

la probabilidad de la palabra w_i, se determina a partir de la relación que define una GIP notada por (G_p), entre la subcadena $w_1 \dots w_k$ y dicha palabra. Esta aproximación presenta varias ventajas. En primer lugar, permite representar de forma compacta y eficiente relaciones a largo término entre las palabras de la cadena; en segundo lugar, existen algoritmos robustos que permiten la evaluación de la expresión (3.3) y posibilitan una integración eficiente de este tipo de modelos. Sin embargo el coste temporal de los algoritmos hace que la utilización se vea limitada. Además, para tareas reales complejas (difíciles) es necesario un elevado número de parámetros y consecuentemente, una gran cantidad de datos para estimarlos adecuadamente.

Estos problemas se acentúan en tareas reales con grandes vocabularios, por lo que la aplicación de las GIP en modelos de lenguaje sin apoyo de algún otro mecanismo resulta poco adecuado [3].

3.2.1. Integración con el modelo de n-gramas

Para aprovechar la fortaleza de cada uno de los modelos se ha propuesto la integración del modelo de n-gramas con modelos estructurales generados por una GIP en un modelo híbrido. Los modelos de n-gramas tratan de representar la información local a nivel de palabras, mientras que los modelos estructurales tratan de representar información sintáctica de toda la frase a

nivel de categorías[4].

Jelinek y Lafferty en [3], plantean el problema del cálculo de (3.3) como:

$$p(w_i \mid w_1 w_2 \dots w_{i-1}, G_p) = \frac{p(S \xrightarrow{*} w_1 \dots w_{i-1} w_i \dots \mid G_p)}{p(S \xrightarrow{*} w_1 \dots w_{i-1} \dots \mid G_p)}. \tag{3.4}$$

Esta aproximación se caracteriza porque para el cómputo de la probabilidad de la siguiente palabra se considera toda la historia anterior, aportando de esta forma la mayor cantidad posible de información. Para computar la expresión (3.4) se presenta el algoritmo LRI que permite calcular $p\left(S \xrightarrow{*} x_1 \dots x_{i-1} \dots \mid G_p\right)$, esto es, la probabilidad de que a partir de S se genere una cadena cuyo prefijo inicial es $x_1 \dots x_{i-1} \dots$

3.2.2. El Algoritmo LRI

Definición **1.** *Sea*[5] G_p *y* x *una cadena de* $L(G_p)$*. Se define la probabilidad de que el no Terminal* $A \in NT$ *derive directamente en el no terminal* $B \in NT$*, terminal más a la izquierda en una de sus reglas como:*

$$R(A \to B) = \sum_{C \in NT} p(A \to BC), \tag{3.5}$$

donde la ecuación (3.5) indica que se están sumando todas las probabilidades de las reglas cuyo antecedente es A y cuyo elemento mas a la izquierda del consecuente sea B, no importando cual sea el elemento a la derecha de B.

Definición **2.** *La probabilidad de que B sea el no terminal más a la izquierda, en cualquier cadena* $\alpha \in \sum \cup NT$ *que se pueda derivar a partir de A, se define como:*

[4]Una categoría es una función φ que a cada palabra w le asigna un conjunto K donde K pertenece a una familia finita de conjuntos $\Im$

[5]G_p una gramática incontextual se encuentra en forma normal de chomsky (FNC) si las reglas de derivación tienen la forma $A \to BC$ o $A \to v$ donde A, B, C pertenecen a NT y v pertenece a $\sum$

$$
\begin{aligned}
T(A \Rightarrow B) = R(A \to B) \ + & \sum_{C_1 \in NT} R(A \to C_1) R(C_1 \to B) \ + \ldots \\
+ & \sum_{C_1, \ldots, C_k \in NT} R(A \to C_1) R(C_1 \to C_2) \ldots R(C_k \to B) \ + \ldots \\
= & \sum_{\alpha \in (NT \cup \Sigma)} p(A \overset{*}{\Rightarrow} B\alpha \mid G_p).
\end{aligned}
\tag{3.6}
$$

(3.6) se interpreta como la suma de todas las probabilidades de que A derive en B seguido de cualquier otro consecuente y dada la gramática G_p.

DEFINICIÓN **3.** *La probabilidad de que BC pueda ser la subcadena inicial de todas las subcadenas derivadas de A, se define como:*

$$T(A \Rightarrow BC) = p(A \to BC) \ + \sum_{D \in NT} T(A \Rightarrow D) p(D \to BC),$$

esto es, la probabilidad de que A derive en BC más la suma de todas las probabilidades de que algún elemento del conjunto de no terminales sea el no terminal más a la izquierda que se pueda derivar de A, multiplicado por la probabilidad de que ese elemento derive en la subcadena BC.

DEFINICIÓN **4.** *La probabilidad de generación de una cadena a partir de A, cuya subcadena inicial sea $x_i \ldots x_j$, se define como:*

$$e(A << i, j) = p(A \overset{*}{\Rightarrow} x_i \ldots x_j \ldots \mid G_p).$$

Algoritmo LRI

Entrada: $x \in L(G_p)$, G_p.

Salida: $e(A << i, i)$, (probabilidad de que a partir del no Terminal A, se genere la subcadena $x_{i\ldots}x_{j\ldots}$ dentro de la cadena x).

i. $e(A << i, i) = p(A \rightarrow x_i) \ + \sum_{B \in NT} T(A \Rightarrow B) p(B \rightarrow x_i)$

ii. $e(A << i, j) = \sum_{B,C \in NT} T(A \Rightarrow BC) \sum_{k=1}^{j-1} e(B < i, k >)$
$e(C << k+1, j) \ i \neq j.$

La suma de que BC sea la subcadena inicial de todas las formas sentenciales de A, para todo B,C pertenecen a NT, multiplicado por las probabilidades de generación de cadenas a partir del elemento más a la izquierda de la derivación, es decir, B cuya cadena inicial sea $x_i \ldots x_k$ con $k = \{1, \ldots, j-1\}$ seguido de la probabilidad de generación de cadenas a partir del elemento a la derecha de la derivación, es decir, C cuya cadena inicial sea $x_{k+1}, \ldots, x_j$. Como se puede ver, el cálculo de las expresiones $T\,(A \rightarrow B)$ y $T\,(A \rightarrow BC)$ se basa en la expresión (3.6) y ésta se evalúa en términos de la expresión (3.5). En [3] se plantea la utilización de álgebra matricial para la evaluación de la expresión (3.6).

Como Jelineck y Lafferty exponen en [3], el cálculo de $p(x_1 \ldots x_k \ldots \mid G_p)$ a partir del algoritmo LRI, se limita a aquellas gramáticas con un número reducido de no terminales, de forma que el cálculo de la matriz inversa sea posible.

3.2.3. Combinación de n-gramas y GIP

En trabajos recientes de ML [6], se han propuesto modelos híbridos que combinan modelos de n-gramas con modelos estructurales para calcular la expresión (3.1). El modelo de n-gramas da cuenta de las relaciones entre las palabras del léxico, donde quedan mejor representadas las restricciones locales. Por su lado el modelo estructural (GIP) según [3], da cuenta de las restricciones entre categorías de palabras, donde quedan mejor representadas

las restricciones a más largo término.
Para comentar el problema relacionado con el tamaño del vocabulario los investigadores han utilizado la clasificación por categorías, una función

$$\varphi : \sum_{w_i \mapsto K_i} \rightarrow \Im$$

que a cada palabra w_i le asigna un conjunto K_i donde las K_i pertenecen a las K clases.
A continuación se presenta la forma en la cual la probabilidad (3.1) se calcula con un modelo híbrido que combina un modelo de n-gramas con una GIP a nivel de categorías, así la expresión (3.1) se calcula como:

$$\begin{aligned} p(w_i \mid w_1 \dots w_{i-1}) = \gamma\ & p(w_i \mid w_{i-n+1}, \dots, w_{i-1}) \\ + \ (1-\gamma)\ & p(w_i \mid g(w_1) \dots g(w_{i-1}), G). \end{aligned} \tag{3.7}$$

El primer sumando de esta expresión es un modelo de n-gramas que recoge las dependencias locales a nivel de palabras. En el segundo sumando, $g()$ es una función de etiquetado que asocia una categoría a cada palabra. El problema de agrupar las palabras en categorías es un problema que aparece ampliamente estudiado en [3]. La expresión $p(w_i \mid g(w_1) \dots g(w_k), G)$ pretende recoger las relaciones estructurales a largo término entre las categorías de la frase. En la expresión (3.7) $0 < \gamma < 1$, es un factor de peso que pondera ambas partes de la expresión. Se simplificará el cálculo de la expresión (3.7), para ello se supone que se dispone de un método para etiquetar las palabras. Esto permite separar el problema del etiquetado del cómputo de $p(w_i \mid g(w_1) \dots g(w_k), G)$. De esta forma el segundo sumando de (3.7) toma la forma:

$$\begin{aligned} p(w_i \mid g(w_1) \dots g(w_{i-1}), G) &= p(w_i \mid g_1 \dots g_{i-1}) \\ &= p(w_i \mid g_i) p(g_i \mid g_1 \dots g_{i-1}) \end{aligned} \tag{3.8}$$

donde a partir de ahora g_i, $1 \leq i \leq k$, denotará la etiqueta asociada a la *i-ésima* palabra. La expresión anterior se interpreta en los siguientes términos: la probabilidad de que ocurra w_i dado que han ocurrido $g_1 \dots g_{i-1}$, se calcula a partir de que ocurra w_i dado que ocurrió g_i, multiplicado por la probabilidad de que ocurra g_i dado que han ocurrido las etiquetas de $g_1 \dots g_{i-1}$.
El valor $p(w_i \mid g_i)$ corresponde a la probabilidad de clasificación de la palabra w_i en la categoría g_i. Para la estimación de los parámetros de los correspondientes modelos es necesario disponer de un conjunto de datos etiquetados en términos de categorías. De esta forma, el valor $p(w_i \mid g_i)$ se estima a partir de las frecuencias del corpus. Para una palabra w del vocabulario, la probabilidad de que esa palabra se clasifique en la categoría g, será:

$$p(w \mid g) = \frac{N(w,g)}{\sum_{w'} N(w',g)}$$

donde $N(w,g)$ es el número de veces que la palabra w ha sido etiquetada con la etiqueta g. La evaluación de la expresión $p(g_i \mid g_1 \dots g_{i-1})$ puede realizarse con el algoritmo LRI, así:

$$p(g_i \mid g_1 \dots g_{i-1}) = p(g_i \mid g_1 \dots g_k, G_p).$$

La expresión (3.7) se aproxima como:

$$\begin{aligned} p(w_i \mid w_1 \dots w_{i-1}) &= \gamma\ p(w_i \mid w_{i-n+1}, \dots, w_{i-1}) \\ &\quad +\ (1-\gamma)\ p(w_i \mid g(w_1) \dots g(w_{i-1}), G) \\ &= \gamma\ p(w_i \mid w_{i-n+1}, \dots, w_{i-1}) \\ &\quad +\ (1-\gamma)\ p(w_i \mid g_i) p(g_i \mid g_1 \dots g_{i-1}, G). \end{aligned} \tag{3.9}$$

Los parámetros γ se estiman mediante el algoritmo Expectation Maximation (EM) y Back-off en [6].

3.3. Combinación de n-gramas con modelos basados en máxima entropía

El concepto básico de entropía en teoría de la información tiene mucho que ver con la incertidumbre que existe en cualquier experimento o señal aleatoria. Es también la cantidad de ruido o desorden que contiene o libera un sistema. De esta forma, podremos hablar de la cantidad de información que lleva una señal. Por otro lado, al aplicar el concepto al lenguaje, la entropía mide la diversidad de un conjunto de datos, entre mayor sea la entropía hay mayor diversidad en el lenguaje y por lo tanto tiene mayor capacidad expresiva. Más formalmente, dada una distribución de probabilidad sobre el conjunto finito χ la entropía se define como:

$$H(p) = -\sum_{x \in \chi} p(x) \left[log\ p(x) \right].$$

3.3.1. El principio de Máxima Entropía

El principio de Máxima Entropía es un marco bastante útil y flexible para el desarrollo de modelos del lenguaje. Bajo este marco se pueden cobijar unidades lingüísticas de diferente naturaleza (palabras, clases, etc.), así como modelos lingüístistos diversos (*n-gramas*, modelos con caché, gramáticas, *triggers*, etc.), todos unificados por un mismo formalismo. Este principio trata de determinar la distribución probabilística p que hace máxima la entropía notada por $H(p)$, haciendo uso solamente de la información de que se dispone, sin hacer suposiciones teóricas sobre la forma u otra característica de la distribución.

En [6] el problema se plantea así:

"Sea X una variable aleatoria discreta que toma los valores x_i para

$i = 1 \ldots n$, de los cuales se ignoran las probabilidades p_i, pero se conoce el valor esperado de cierta función:

$$E[f(x)] = \sum_{i=1}^{n} p_i f(x_i)$$

Será posible determinar el valor esperado de la función $f(x)$?"

La información proporcionada no es suficiente, en [6] usando el concepto de entropía se propone la siguiente solución:

"Al hacer inferencia con base en información parcial, se debe utilizar la distribución de probabilidades que tenga máxima entropía sujeta a todo aquello que es conocido "

Esta solución coincide con la definición de máxima entropía que se conoce actualmente. Para comprender mejor el principio de máxima entropía (PME) supóngase que se quiere hacer inferencia sobre la distribución de probabilidades de las frases en un lenguaje determinado. La evidencia de que se dispone para realizar tal inferencia, es decir los datos conocidos, es el corpus sobre el cual se está basando el estudio. De acuerdo con el PME, se debe encontrar la distrubución de probabilidades con máxima entropía sobre las frases del lenguaje y que esté sujeta a las restricciones de la información contenida en el corpus. Por ejemplo, la experiencia ha demostrado que los trigramas son piezas de informacíon importantes; entonces,se puede incluir en el modelo características que representan a los trigramas . Las funciones encargadas de medir la ocurrencia de los trigramas deben cumplir las restricciones impuestas por el modelo.

El problema de maximizar la entropía: Entre todas las distribuciones de probabilidad que pertenecen a p_i se trata de encontar aquella que haga máxima la entropía [6].

3.3.2. Divergencia de Kullback-Leibler

Dado un conjunto finito χ, y dos distribuciones de probabilidad p,q sobre χ, se define la divergencia [6] entre p y q como:

$$D(p \mid\mid q) = \sum_{x \in \chi} p(x) log \frac{p(x)}{q(x)}, \tag{3.10}$$

se puede demostrar que $D(p \mid\mid q)$ cumple las siguientes propiedades:

1. $D(p \mid\mid q)$ es una función no negativa de valor real.
2. $D(p \mid\mid q) = 0$ si y solamente si $p = q$.
3. $D(p \mid\mid q)$ es estrictamente convexa en p y q separadamente.
4. $D(p \mid\mid q)$ es continuamente diferenciable en q.

Ahora bien

$$\begin{aligned} D(p \mid\mid q) &= \sum_{x} p(x) log \frac{p(x)}{q(x)} \\ &= \sum_{x \in \chi} p(x) \left[log\ p(x) - log\ q(x)\right] \\ &= \sum_{x \in \chi} p(x) \left[log\ p(x)\right] - \sum_{x} p(x) \left[log\ q(x)\right] \\ &= -H(p) - \sum_{x \in \chi} p(x) \left[log\ q(x)\right], \end{aligned}$$

donde $H(p)$ es la entropía de p. Si la distribución q es la distribución uniforme, $q(x) = \frac{1}{|\chi|}$ para todo $x \in \chi$, se tiene:

$$D(p \mid\mid q) = -H(p) + R,$$

donde R es una constante. Entonces el problema de hallar un máximo para la entropía es equivalente a hallar un mínimo para la divergencia res-

[6] También denominada entropía relativa de p respecto a q. Sólo se trata el caso en que X es finito.

pecto a la distribución uniforme. Bajo este marco el problema de hallar la distribución de máxima entropía relativa a la distribución q, se plantea como sigue:

"Dada la familia lineal ζ de distribuciones de probabilidad sobre el conjunto finito χ

$$\zeta = \left\{ p : \sum_{x \in \chi} p(x) f_i(x) = K_i \right\}; \tag{3.11}$$

para $i = 1, \ldots, m$, una distribución de probabilidades q denominada distribuciòn a priori y un conjunto f_i de funciones denominadas funciones características. La distrubución de probabilidades $\hat{p} \in \zeta$ que haga mínima la divergencia respecto a q se denomina distribución de mínima divergencia o de máxima entropía[7] *relativa a q: "*

$$\hat{p} = Arg \left\{ \min_{p \in \zeta} D(p \mid\mid q) \right\}. \tag{3.12}$$

Según las propiedades, D es no negativa y estrictamente convexa, lo cual garantiza la existencia de $\hat{p}$ y además puede demostrarse que es la única solución de mínima divergencia [6].

Obsérvese que se trata de un problema de optimización sujeto a las restricciones. El método natural utilizado para resolver tal problema es el de los multiplicadores de Lagrange. El Lagrangiano está dado por la expresión:

$$\begin{aligned} \wedge (p, \lambda) &= D(p \mid\mid q) - \sum_i \lambda_i \left[\sum_{x \in \chi} p(x) f_i(x) - K_i(x) \right] \\ &= \sum_{x \in \chi} p(x) log \frac{p(x)}{q(x)} - \sum_i \lambda_i \left[\sum_{x \in \chi} p(x) f_i(x) - K_i(x) \right], \end{aligned} \tag{3.13}$$

[7] Cuando la distribución q es la distrubición uniforme, $\hat{p}$ se denomina simplemente distribución de máxima entropía.

donde $\lambda = \{\lambda_1, \ldots, \lambda_n\}$. Derivando la expresión en (3.13) con respecto a p e igualando a cero se obtiene finalmente la distribución exponencial

$$\hat{p}(x) = \frac{1}{Z}\ q(x)\ exp\left\{\sum_{i=1}^{n} \lambda_i f_i(x)\right\}, \tag{3.14}$$

donde Z es la constante de normalización de la distribución y los λ_i, que son los multiplicadores de lagrange, son los parámetros que representan los pesos de aporte de cada una de las características al modelo. Las f_i serán las que determinen y ponderen la ocurrencia de cada característica dentro de los eventos concretos. la distribución q es denominada distribución *apriori* y fija la forma inicial de la distribución $\hat{p}$.

En (3.14) se ha obtenido la forma de la distribución de máxima entropía, sin embargo los parámetros λ_i son desconocidos, es decir en el proceso hasta ahora solamente se ha optimizado respecto a p y falta hacerlo respecto a λ. Sustituyendo en (3.13) el p dado por (3.14) y K_i por $E_{\tilde{p}[f_i]}$ se tiene:

$$\begin{aligned} \wedge(p, \lambda) = &\sum_x p(x)\,[log\ p(x)] - \sum_x p(x)\,[log\ q(x)] - \\ &\sum_{i=1}^{n}\sum_x \lambda_i p(x) f_i(x) + \sum_{i=1}^{n} \lambda_i E_{\tilde{p}[f_i]} \\ \wedge(p, \lambda) = &-\log Z + \sum_{i=1}^{n} \lambda_i E_{\tilde{p}[f_i].} \end{aligned} \tag{3.15}$$

Ahora se tienen que encontrar los valores de λ_i que maximizan la función mostrada en (3.15). Si se define $\varphi(\lambda) = \wedge(p, \lambda)$ en resumen el problema que falta por resolver es encontrar

$$\hat{\lambda} = Arg\left\{\max_{\lambda} \varphi(\lambda)\right\}$$

Desde el punto de vista estadístico la función de verosimilitud es la función más usada para encontrar los estimadores de los parámetros de las diferentes

funciones de probabilidad, por ese motivo se utilizará la verosimilitud respecto a la distribución empírica $\tilde{p}$ y está dada por: $\prod p(x)^{\tilde{p}(x)}$ y por tanto el logaritmo de la verosimilitud que se notará $L_{\tilde{p}(\lambda)}$, tiene la forma:

$$\begin{aligned} L_{\tilde{p}}(\lambda) &= \sum_{x\in\chi} \tilde{p}(x) log\ p(x) \\ &= \sum_{x\in\chi} \tilde{p}(x) \left\{ log\ q(x) + \sum_{i=1}^{n} \lambda_i f_i(x) - log\ Z \right\} \\ &= -log\ Z + \sum_{i=1}^{n} \lambda_i f_i(x) E_p\left[f_i\right] + \sum_{x\in\chi} \tilde{p}(x) log\ q(x). \end{aligned} \tag{3.16}$$

Comparando (3.15) y (3.16) se concluye que maximizar $\varphi(\lambda)$ es equivalente a maximizar $L_{\tilde{p}}(\lambda)$. De esa manera los parámetros del modelo se obtienen encontrando el valor de λ que haga máxima la verosimiltud[8].

3.3.3. Modelo de lenguaje de máxima entropía

Una de las posibles aplicaciones del PME está en el modelado de lenguaje, usando el PME se puede reunir en el mismo marco información de diferente naturaleza, lo que no ocurre con los modelos clásicos.

Desde el punto de vista del modelado de lenguaje se puede pensar que el dominio de definición de la distribución de máxima entropía es el conjunto W^+ que, como se ha definido, es el conjunto de frases $w = w_1 \dots w_n$; $w_i \in W$ donde W es un vocabulario dado. La información disponible es tratada en términos de características, propiedades medibles de la frase. Las funciones f_i en este caso son funciones unitarias que establecen si determinada característica es satisfecha o no por la frase $w \in W^+$. El conjunto sobre el cual se define la distribucion de mínima divergencia es finito; al considerar el con-

[8]En [5] se muestran algunos algoritmos para la estimación de los parámetros.

junto W^+ se estaría utilizando como conjunto de definición de la distribución a un conjunto infinito.
Sea Ω un corpus de datos, K_i las constantes mencionadas en (3.11), usualmente $K_i = E_{\tilde{p}}[f_i]$ donde $E_{\tilde{p}}[f_i]$ son los valores esperados empíricos de f_i sobre el corpus Ω. Entonces la familia lineal sobre la cual se busca la distribución de mínima divergencia respecto a q es:

$$\zeta = \left\{p : \sum_{w \in W^+} p(w) f_i(w) = E_{\tilde{p}}[f_i]\right\};$$

para $i = 1, \ldots, m$. Dicho en otros términos, se trata de encontrar la distribución de probabilidades $\tilde{p}$ respecto a q que haga mínima la divergencia sobre las cadenas de $W^l \subset W^+$ donde W^l es el conjunto de todas las cadenas de longitud l y que cumpla con las restricciones:

$$\sum_{w \in W^l} p(w) f_i(w) = \sum_{w \in \Omega} \tilde{p}(w) f_i(w) \text{ para } i = 1, \ldots, m$$

3.3.4. Modelo de lenguaje condicional de máxima entropía

En el modelo condicional de máxima entropía (MEC) el cálculo de probabilidades de una frase se lleva a cabo utilizando la aproximación

$$p(w) \approx \prod_{i=1}^{n} p(w_i \mid \phi(h_i))$$

vista en el capítulo 2, de manera que el problema se reduce a determinar una distribución de probabilidades de la forma $p(w_i \mid \phi(w_1 \ldots, w_{i-1}))$. Usualmente en los MEC el conjunto W^l se reduce (por una relación de equivalen-

cia) al espacio de eventos $\chi \times W$, donde χ se denomina el contexto[9] y W es el vocabulario. Sobre este espacio de eventos está definida la distribución conjunta $p(x, y)$ donde $x \in \chi$ y $y \in W$.
El espacio de eventos χ más elemental es el denominado *modelo Markoviano* de orden n en el cual $\chi = W^n$: el conjunto de las cadenas de longitud n. Teniendo en χ como contexto el modelo Markoviano de orden n, se han desarrollado los modelos de $n - gramas$ de máxima entropía [5].
Teniendo en cuenta que $p(x, y) = p(x)p(y \mid x)$ y utilizando los mismos argumentos expuestos hasta ahora, la distribución de máxima entropía condicional $p(y \mid x)$, cuya formulación está dada por la expresión:

$$p(y \mid x) = \frac{1}{Z(x)} exp \left\{ \sum_{i=1}^{n} \lambda_i f_i(x, y) \right\}$$

donde $Z(x) = \sum_y exp \{\sum_{i=1}^{n} \lambda_i f_i(x, y)\}$ es la constante de normalización, que depende del contexto. Los λ_i son los parámetros del modelo y las f_i son las funciones indicatrices.

Suponiendo que el contexto es un modelo de Markov de orden 2, es decir $\chi x W = \{w_1 w_2 \mid w_i\}$, donde interesa incluir en el modelo la información aportada por los bigramas, entonces se define una función indicatriz $f_{a,b}$ como sigue:

$$f_{\{a,b\}}(x, y) = \begin{cases} 1 & si\ x\ termina\ en\ a\ y\ y = b \\ 0 & en\ otro\ caso \end{cases}$$

[9]Por contexto se entiende el conjunto de elementos de información que le aportan al modelo.

con restricción asociada:

$$\sum_h \tilde{p}(y) \sum_w p(x|y) k_{\{a,b\}}(x,y) = \tilde{E}[f_{\{a,b\}}]$$

Usando triggers la respectiva función $f_{A \to B}(x,y)$ se define como:

$$k_{A \to B}(x,y) = \begin{cases} 1 & si\ A \in x, y = B \\ 0 & en\ otro\ caso \end{cases}$$

y la respectiva restricción esta dada por:

$$\sum_h \tilde{p}(x) \sum_w p(y|x) k_{A \to B}(x,y) = \tilde{E}[k_{A \to B}]$$

3.3.5. Ventajas y desventajas del MEC

Los aspectos más sobresalientes de los modelos MEC son:

- Su formulación es simple y general, además utiliza simplemente la información que tiene disponible; no hace ninguna suposición sobre el modelo.
- Tiene la capacidad de incluir información de diferente naturaleza en un mismo marco.

Presenta ciertas limitaciones entre las que se destacan las siguientes:

- El coste computacional del entrenamiento en tareas de alguna complejidad es elevado, principalmente en el cálculo de la constante de normalización $Z(x)$, que ha de ser calculada para toda historia en cada iteración.
- Hay características contenidas en una frase que no es posible modelar correctamente con los modelos condicionales, por ejemplo su longitud, la coherencia semántica o las construcciones sintácticas.

- Los modelos condicionales asumen independencia al momento de calcular la probabilidad de una frase, es decir: en los modelos condicionales se aproxima $p(w_i \mid w_1, \ldots, w_{i-1})$ mediante $p(w_i \mid \phi(h_i))$ asumiendo que la ocurrencia de w_i es independiente de la ocurrencia de algunos de los w_k anteriores.

Capítulo 4

Resultados Experimentales

4.1. Introducción

En este capítulo se presentan experimentos realizados con los modelos de lenguaje de n-gramas y se estudia el efecto de los suavizados sobre la calidad del modelo. Estos resultados se han obtenido usando un corpus notado Ω el cual se dividió en dos conjuntos disjuntos, uno de entrenamiento (E) y otro de prueba (P) o de validación; la partición de estos conjuntos se hace de tal forma que (E) y (P) equivalen al 80 y 20 por ciento de Ω, respectivamente. Con el conjunto de entrenamiento se estima el modelo y con el de prueba se estudia la calidad del modelo utilizando la perplejidad, sin embargo, para efectos de comparación, también se utilizaron los datos de entrenamiento para determinar la calidad del modelo.

4.2. Marco experimental

En esta sección se describen las condiciones en las cuales se realizaron los experimentos y las herramientas utilizadas para su ejecución. El objetivo de los experimentos es determinar cuál técnica de suavizado produce modelos de mejor calidad; la calidad del modelo se mide en su capacidad expresiva

es decir, que medir las probabilidades asignadas reflejen la frecuencia de aparición de las frases en la realidad.

4.2.1. Corpus de datos y corpus de prueba

El corpus de datos utilizado es una parte del corpus denominado *Wall Street Journal* (WSJ), procesado en el proyecto Penn TreeBank [6]. Se decidió hacer uso de esta base porque es ampliamente conocida y divulgada.
El WSJ es una colección de textos de ediciones de finales de los 80 del periódico *Wall Street Journal*, este corpus está analizado y etiquetado automáticamente y revisado de forma manual. El corpus está agrupado en 25 directorios (del 00 al 24) de ellos se seleccionaron 12 aleatoriamente y se formó un conjunto de frases que se dividió en dos: un conjunto de entrenamiento con 4921 frases y un conjunto de prueba con 1272 frases. A continuación se muestra como ejemplo, la frase "Pierre Vinken, 61 years old, will join the board as nonexecutive director Nov. 29."que fué analizada y etiquetada en el proyecto Penn TreeBank.

```
((S
(NP-SBJ
   (NP(NNP  Pierre)(NNP  Vinken))
   (,  ,)
   (ADJP
      (NP(CD  61)(NNS  years))
      (JJ  old))
   (,  ,))
(VP(MD  will)
   (VP(VB  join)
      (NP(DT  the)(NN  board))
      (PP-CLR(IN  as)
         (NP(DT  a)(JJ  nonexecutive)(NN  director)))
```

(NP-TMP (NNP Nov.)(CD 29))))
(. .)))

4.2.2. Metodología experimental

Usando el mismo conjunto de entrenamiento se estima un modelo de trigrama por cada suavizado el cual es obtenido a partir de la herramienta de software descrita en (4.2.3) y se pretende analizar los resultados arrojados por los modelos.

4.2.3. Hardware y software

Los experimentos se realizan utilizando la herramienta de software *CMU-Cam - Toolkit - v2*, que fue desarrollada por Ronald Rosenfeld y Philip Clarkson[10] en el año de 1996, es de dominio público y se ejecuta bajo las plataformas Linux y Unix.
Para los experimentos del presente trabajo se utilizó la versión ubuntu 9.10, linux 2.6.31-14, un computador personal Dell de referencia 1525 con procesador intel core duo y memoria RAM de 3 GB.

4.2.4. Descripción de los experimentos

En los presentes experimentos se van a obtener modelos de trigramas utilizando cada una de las técnicas de suavizado discutidas en el capítulo 2. Se evaluará la calidad del modelo [11] y se compararán los resultados para determinar cuál suavizado es más efectivo en el sentido de producir modelos de mejor calidad. A continuación se describe el proceso mediante el cual se

[10] *CMU-Cam-Toolkit-v2* descargado del sitio web *www.speech.cs.cmu.edu/SLM/toolkit.html* en Octubre del año 2009.

[11] La calidad del modelo se medirá utilizando la perplejidad, concepto que se definirá en la siguiente subsección.

obtiene el modelo de trigramas utilizando el corpus de datos que se tiene como base.
De la herramienta de software *CMU-Cam-Toolkit-v2* se usan los programas

text2wfreq: realiza un listado de cada una de las palabras presentes en el corpus de datos y encuentra el número de veces que estas ocurren.

wfreq2vocab: crea un vocabulario el cual es generado teniendo en cuenta los datos arrojados por ***text2wfreq***.

text2idngram: hace un listado del número de trigramas que ocurren en el texto con su respectiva frecuencia.
La distribución de probabilidades en el modelo de n-gramas está basada en la distribución condicional de los unigramas, los bigramas y los trigramas y viene dada por:

$$p(w_3 \mid w_1, w_2) = \begin{cases} p_3(w_1, w_2, w_3) & si\ (w_1, w_2, w_3)\quad existe \\ \alpha_2(w_1, w_2)p(w_3 \mid w_2) & si\ (w_1, w_2)\ existe \\ p(w_3 \mid w_2) & en\ otro\ caso. \end{cases}$$

$$p(w_2 \mid w_1) = \begin{cases} p_2(w_1, w_2) & si\ (w_1, w_2)\ existe \\ \alpha_1(w_1)p_1(w_2) & en\ otro\ caso, \end{cases}$$

donde $p_3(w_1, w_2, w_3)$ es el cociente entre el número de veces que se observa el trigrama (w_1, w_2, w_3) y el número total de trigramas. De igual forma $p_2(w_1, w_2)$ es el cociente entre el número de veces que se observa el bigrama (w_1, w_2) sobre el número total de bigramas, α_i es un parámetro de discontinuidad que depende de modelo al cuál se refiera (unigrama, bigrama o trigrama). Esta fórmula de distribución es la misma para todos los modelos

que se están considerando.

idngram2lm: desarrolla un modelo de lenguaje con las técnicas de suavizados, siendo este diferente para cada una de ellas. El ***idngram2lm*** muestra cada trigrama con el logaritmo en base 10 de cada probabilidad, además da información acerca de los bigramas y los unigramas.

evallm: encuentra la perplejidad de un modelo de lenguaje respecto a un corpus permitiendo ver cual técnica de suavizado produce modelos más eficientes.
Las técnicas de suavizado usadas para generar los modelos son las siguientes:

1. Técnica de suavizado lineal.
2. Técnica de suavizado Reducción absoluta.
3. Técnica de suavizado de Good Turing.
4. Técnica de suavizado de Witten Bell.

Se realizaron diferentes particiones del corpus y en distintos porcentajes con el fin de medir la calidad del modelo sin que importe la forma de tomar los conjuntos de entrenamiento y prueba en un mismo corpus. Los porcentajes manejados son.

1. 90 por ciento conjunto de entrenamiento, 10 por ciento conjunto prueba.
2. 80 por ciento conjunto de entrenamiento, 20 por ciento conjunto prueba.
3. 70 por ciento conjunto de entrenamiento, 30 por ciento conjunto prueba.

Cada partición se tomó al comienzo, en la mitad y al final del corpus, es decir, para el caso 1 se tomó el 90 por ciento que equivale al conjunto de entrenamiento al comienzo del corpus y el 10 que es el conjunto de prueba al

final del corpus, sin embargo como se verá en la sección 4.2.6 al realizar estas particiones los valores que arrojan los modelos con estas particiones no tienen mayor diferencia, esto se presenta debido a que las frases han sido escogidas al azar, luego los resultados serán similares independiente de donde se tomen las particiones en el corpus. Todas las particiones se hicieron manualmente.

4.2.5. Calidad del modelo

La calidad del modelo se medirá mediante la perplejidad sobre el conjunto de prueba P; la perplejidad notada PP se define como:

$$PP = \exp\left(\frac{-\sum_{x\in\Omega}\log(p(x))}{\sum_{x\in\Omega}|x|}\right).$$

Esta es una medida utilizada en la teoría de la información y mide la capacidad del modelo para representar el lenguaje; la perplejidad desde el punto de vista del modelado de lenguaje, es la que mide la proporción de frases que hay en el lenguaje. Un modelo es mejor cuando es más expresivo en el sentido que represente mejor el lenguaje que se está modelando, el modelo más expresivo es aquel que tiene menor perplejidad.

La perplejidad puede interpretarse mas o menos como la media geométrica del lenguaje: una lengua con perplejidad X tiene aproximadamente la misma dificultad que otro lenguaje en el cual cada palabra es seguida por X diferentes palabras con probabilidades iguales.
La perplejidad es una función tanto para un modelo como para un texto. Este hecho deberá tenerse en cuenta cuando se comparan los números de perplejidad para diferentes textos y diferentes modelos. Una comparación significativa se puede realizar entre perplejidades de varios modelos, todos con respecto al mismo texto y el mismo vocabulario. El tamaño del vocabulario debe ser el mismo, o bien cuando el tamaño del vocabulario es más pequeño, paradójicamente, arrojará el modelo de mas baja perplejidad (ya que normalmente excluye palabras raras).

La perplejidad mide la diversidad de un lenguaje, se relaciona con la entropía que se presenta en un lenguaje, así entre mayor entropia, menor perplejidad y se puede ver también como $2^{-H(x)}$ donde $H(x)$ es la entropía definida como:

$$H(x) = -\sum p(x)\log(p(x)).$$

Esta mide la diversidad de un conjunto de datos, es decir, entre mayor sea la entropía hay mas diversidad en el lenguaje y por lo tanto tiene mayor capacidad expresiva, así entre mayor entropía, menor perplejidad.

4.2.6. Resultados

En esta sección se presentan los resultados obtenidos mediante los experimentos realizados anteriormente, como éstos son muy parecidos puesto que se está trabajando con un mismo corpus, sólo se mencionarán los resultados para las particiones de 80, 20 por ciento y 90 y 10 por ciento.
Para la partición de 80 por ciento conjunto de entrenamiento, 20 por ciento conjunto prueba, los resultados son:
Tamaño de vocabulario: 11857 palabras
Frecuencia de las palabras en el corpus: 119958
Número de trigramas obtenidos: 97 197
Número de bigramas obtenidos: 61 142
Número de unigramas obtenidos: 11 857
En la Tabla 1 se muestran los resultados de las perplejidades en los datos analizados.
Partición de 80 y 20 por ciento.

Suavizados	**Corpus de datos**	
	PP Datos de entrenamiento	**PP Datos de prueba**
Lineal	11.52	266.09
Reducción Absoluta	16.46	220.90
Good Turing	16.64	218.56
Witten Bell	5.34	237.96

Tabla 1. Resultados experimentales tomando el 80 por ciento datos de entrenamiento y el 20 por ciento datos de prueba.

1. Se observa la diferencia que hay en el resultado de la perplejidad cuando se usan los datos de entrenamiento y los datos de prueba. La perplejidad en todos los casos en los que se utilizan los datos de entrenamiento es mucho menor que la perplejidad cuando se utilizan los datos de prueba, esto es debido a que con E se entrenó el modelo, por tanto la mayor cantidad de datos eran conocidos y las probabilidades son frecuencias relativas y por eso están más atadas a los datos y así se produce una baja perplejidad; por el contrario, P no ha sido visto por el modelo lo cual le presenta mayor dificultad a este para entenderlo. Se debe tener en cuenta que la perplejidad se calcula con los datos de prueba, sin embargo en el presente experimento se ha querido mostrar lo que sucede con ambos conjuntos para efectos de comparación.

2. Al observar lo que sucede con los datos de entrenamiento, la técnica de suavizado de Witten Bell es la que produce mejores modelos, eso podría significar que es el que representa mejor el modelo, sin embargo al comparar las técnicas de suavizado con los datos de prueba se observa que esta técnica no es la mejor; aquí la técnica más efectiva es la de Good Turing y en realidad es la que mejores resultados ha arrojado según los experimentos realizados por otros autores.

3. De los modelos, el de mayor perplejidad es el obtenido con la técnica de suavizado lineal, que como se había dicho es el suavizado más simple y el que genera peores modelos; recuérdese que ésta técnica consiste en sumar una constante a cada una de las frecuencias de los trigramas.

Para la partición de 10 por ciento conjunto de entrenamiento, 90 por ciento conjunto prueba, a continuación se presentan resultados:
Tamaño de vocabulario: 11857 palabras
Frecuencia de las palabras en el corpus: 119958
Numero de trigramas obtenidos: 97 197
Numero de bigramas obtenidos: 61 142
Numero de unigramas obtenidos: 11 857

Suavizados	**Corpus de datos**
	PP Datos de prueba
Lineal	282.79
Reducción Absoluta	225.61
Good Turing	226.79
Witten Bell	238.95

Tabla 2. Resultados experimentales tomando el 90 por ciento datos de entrenamiento y el 10 por ciento datos de prueba.

Los resultados presentados en la Tabla 2 son muy cercanos a los presentados en la tabla 1, el cambio en estos valores se debe a que el conjunto de datos de entrenamiento es mas pequeño luego el modelo va a conocer menos datos por tanto la perplejidad aumenta muy poco con respecto a la obtenida en la tabla 1, esto quiere decir que en un mismo corpus los datos que se obtiene no dependen de la forma como se realizen las particiones.

Capítulo 5

Conclusiones y trabajos futuros

5.1. Conclusiones

1. Computacionalmente el modelo de n-gramas es el más eficiente y el más sencillo de implementar. Experimentos similares al realizado en este trabajo con modelos gramaticales y modelos de máxima entropía han mostrado mayor complejidad temporal y espacial [6].

2. Experimentos realizados muestran que de todas las técnicas de suavizado analizadas, la más eficiente es la técnica de Good-Turing, esto se evidencia con los resultados de la perplejidad obtenidos en la Tabla 1.

3. Se hizo un análisis completo del modelo de n-gramas, ventajas y desventajas y la solución de algunos inconvenientes que éste presenta.

5.2. Trabajos futuros

1. Estudiar otras variaciones del modelo de n-gramas, es decir para $n > 4$ y observar que ocurre en estos casos.

2. Realizar la implementación en un lenguaje de alto nivel.

3. Desarrollar un estudio empírico de algunos métodos para combinar el modelo de n-gramas con otros modelos.

Bibliografía

[1] S. Chen and J. Goodman. *An Empirical Study of Smoothing Techniques for Language Modeling*. Center for Research in Computing Technology, Harvard University, **TR-10-98** (1998).

[2] L.R. Bahl and P.F. Brown and P. V. de Souza and R. L. Mercer. *A tree-based statistical language model for natural language speech recognition*. ITASSP, **37** (1989), 1001–1008.

[3] J. A. Sánchez. *Estimación de gramáticas incontextuales probabilísticas y su aplicación en modelización del lenguaje*. Departamento de Sistemas Informáticos y Computación, Univeridad Politécnica de Valencia, Tesis Doctoral, (1999).

[4] L.R. Bahal and F.Jelinek, and R. L. Mercer. *A Maximun Likelihood Approach to Continuous Speech Recognition*. IEEE Trans. on Pattern analysis and Machine Intelligence, **5** (1983), 179–190.

[5] F. Jelinek. *Up from Trigrams! The Struggle for Improved Language Models*. EUROSPEECH, **3**, (1998), 1034–1040.

[6] F. Amaya. *Algunos aportes a los modelos de lenguaje de máxima entropía de frase completa*. Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, Tesis doctoral, (2001).