

STATISTICAL STRUCTURED PREDICTION

Question set (Part 1-B)

Alumno: Raúl García Crespo

Question 1.

Si las probabilidades iniciales de las reglas son equiprobables y por lo tanto todas las reglas con la misma parte izquierda poseen la misma probabilidad, siempre manteniendo que dichas probabilidades sumen 1, entonces todas las posibles frases generadas por estas reglas que posean la misma estructura (la misma derivación en cuanto a reglas que generan no terminales) poseerán también la misma probabilidad de ser generadas por la gramática, independientemente de los elementos terminales que las componen. Esto sucede porque al ser las probabilidades equiprobables, todos los posibles terminales generados por una regla tienen la misma probabilidad independientemente del terminal que se genera, lo que en el ejemplo que aparece en las transparencias de **Part 1.4 Model Parameter Estimation** provoca por ejemplo que todos los posibles nombres se generen con una probabilidad de 0.2 ya que existen 5 posibles nombres.

Durante la estimación de parámetros mediante el algoritmo Inside-Outside, esto se traduce en que para árboles de derivación con la misma estructura de no-terminales, solamente será necesario calcular la probabilidad de emisión de ese árbol una vez; esto ocurre porque independientemente de los terminales la probabilidad será la misma. Como ejemplo de este comportamiento se emplean las frases: {"la mujer vieja ayuda", "la mujer oculta pelea"}. Ambas frases están formadas por un conjunto de terminales de la gramática diferentes pero poseen un árbol de derivación en común con la misma estructura de no-terminales, por lo que su probabilidad inicial en el algoritmo Inside-Outside será la misma.

Probabilidades de emisión con reglas equiprobables:

$$P_{\theta}(\text{la mujer vieja ayuda}) = 1.0 * 0.33 * 1.0 * 0.2 * 0.5 * 0.5 * 0.25$$

$$\begin{aligned} P_{\theta}(\text{la mujer oculta pelea}) \\ = 1.0 * 0.33 * 1.0 * 0.2 * 0.5 * 0.5 * 0.25 + 1.0 * 0.33 * 1.0 * 0.2 \\ * 0.5 * 0.25 * 0.2 \end{aligned}$$

Para la primera frase y la segunda frase existe un posible árbol de derivación para la estructura sintáctica (Art + Nom + Adj + Verb). Cualquier otra frase con esta misma estructura tendrá la misma probabilidad para ese árbol de derivación concreto en la primera iteración del algoritmo Inside-Outside, lo que se traduce en que solo es necesario calcular la probabilidad una vez para todas las frases con esta estructura.

Question 2.

Todo lo descrito en el ejercicio anterior se aplica también al algoritmo Viterbi.

Question 3.

Al incluir una restricción mediante paréntesis sobre la muestra de entrenamiento “la vieja demanda ayuda” con respecto a la versión que podemos encontrar en las transparencias de **Part 1.4 Model Parameter Estimation**, esta muestra de entrenamiento pierde un posible árbol de derivación. Esto se debe a que la derivación que incluye la regla (Suj → Art Adj Nom) que aparece en la segunda derivación posible para esta muestra en el ejemplo ahora incumple la restricción impuesta por los paréntesis al separar el segmento “demanda ayuda” en dos subárboles diferentes.

Por lo tanto la probabilidad de emisión de esta muestra por parte de la gramática se ve afectada, mientras que el resto de muestras mantienen sus probabilidades:

$$P_{\theta}((la\ vieja)(demanda\ ayuda)) = 0.0009$$

$$P_{\theta}(La\ mujer\ oculta\ pelea) = 0.0009 + 0.01176 = 0.01266$$

$$P_{\theta}(La\ vieja\ ayuda) = 0.007$$

Y con estas probabilidades y la fórmula para la estimación de probabilidades del algoritmo Inside-Outside:

$$p(A \rightarrow \alpha) = \frac{\sum_{x \in D} \frac{1}{P_{\theta}(x)} \sum_{t_x \in x} N(A \rightarrow \alpha, t_x) * P_{\theta}(x, t_x)}{\sum_{x \in D} \frac{1}{P_{\theta}(x)} \sum_{t_x \in x} N(A, t_x) * P_{\theta}(x, t_x)}$$

La probabilidad de la regla (Suj → Art Nom Adj) es:

$$\begin{aligned} p(Suj \rightarrow Art\ Nom\ Adj) &= \\ &= \frac{\frac{1}{0.01266} \cdot 0.01176}{\frac{1}{0.0009} \cdot 0.0009 + \frac{1}{0.01266} \cdot (0.0009 + 0.01176) + \frac{1}{0.007} \cdot 0.007} \\ &= \frac{0.9289}{3} = 0.3096 \end{aligned}$$

Como se puede observar, este resultado no varía con respecto al ejemplo dado que el nuevo valor de la muestra ((la vieja)(demanda ayuda)) solo interviene en el denominador de la expresión, donde su valor se anula con el de su único árbol de derivación pasando a ser 1.

En cambio el resto de reglas con el no terminal “Suj” en su parte izquierda si se ven afectadas por las restricciones de los paréntesis. Por ejemplo para la regla (Suj → Art Nom) su probabilidad será:

$$p(Suj \rightarrow Art Nom) =$$

$$= \frac{\frac{1}{0.007} \cdot 0.007 + \frac{1}{0.01266} \cdot 0.0009 + \frac{1}{0.0009} \cdot 0.0009}{\frac{1}{0.0009} \cdot 0.0009 + \frac{1}{0.01266} \cdot (0.0009 + 0.01176) + \frac{1}{0.007} \cdot 0.007} = \frac{2.071}{3} = 0.6936$$

Y para la regla (Suj → Art Adj Nom) la probabilidad será:

$$p(Suj \rightarrow Art Adj Nom) = \frac{0}{3} = 0$$

Ya que esta última regla desaparece de las posibles derivaciones de las muestras de entrenamiento cuando aplicamos las restricciones de los parámetros. Si sumamos las probabilidades de las reglas (Suj → Art Nom) y (Suj → Art Nom Adj) podemos ver que el resultado es 1, ya que ahora toda la probabilidad de derivar desde el no terminal “Suj” se reparte entre estas dos reglas.

Question 4.

Partiendo de las muestras de entrenamiento y sus posibles derivaciones del aparatado anterior se va a calcular de nuevo la probabilidad de las reglas (Suj → Art Adj Nom) , (Suj → Art Nom) y (Suj → Art Nom Adj) mediante Viterbi. En este caso la fórmula para la estimación de las probabilidades de las normas es la siguiente:

$$\bar{p}(A \rightarrow \alpha) = \frac{\sum_{x \in \mathcal{D}} N(A \rightarrow \alpha, \hat{t}_x)}{\sum_{x \in \mathcal{D}} N(A, \hat{t}_x)}.$$

En este caso solamente se tienen en cuenta las derivaciones más probables y el número de apariciones de la regla y su parte izquierda en dichas derivaciones:

$$p(Suj \rightarrow Art Nom Adj) = \frac{1}{3} = 0.333$$

$$p(Suj \rightarrow Art Nom) = \frac{2}{3} = 0.666$$

$$p(Suj \rightarrow Art Adj Nom) = \frac{0}{3} = 0$$

Ya que al igual que en el ejercicio anterior, la regla (Suj → Art Adj Nom) ya no aparece en ninguna derivación tras las restricciones impuestas por los paréntesis sobre la primera muestra de entrenamiento.

Question 5.

En este ejercicio debemos calcular de nuevo la probabilidad de la regla (Suj → Art Nom Adj) empleando las reglas y probabilidades del ejemplo de las transparencias pero partiendo de un conjunto de entrenamiento diferente. Dicho conjunto de entrenamiento es el siguiente: {la vieja demanda ayuda, la mujer oculta pelea, la vieja mujer oculta demanda ayuda}. Sin embargo con las reglas disponibles en el ejemplo, la gramática no sería capaz de generar la tercera frase del conjunto de entrenamiento, por lo que será necesario incluir por lo menos una nueva regla para que esto suceda.

Se ha optado por incluir la regla (Suj → Art Adj Nom Adj), por lo que será necesario asignarle una probabilidad inicial así como modificar las probabilidades del resto de reglas que posean “Suj” en su parte izquierda para garantizar que estas sigan sumando 1. Tras las modificaciones las reglas afectadas quedarían con las siguientes probabilidades iniciales:

$$p(\text{Suj} \rightarrow \text{Art Nom}) = 0.4$$

$$p(\text{Suj} \rightarrow \text{Art Adj Nom}) = 0.3$$

$$p(\text{Suj} \rightarrow \text{Art Nom Adj}) = 0.2$$

$$p(\text{Suj} \rightarrow \text{Art Adj Nom Adj}) = 0.1$$

Con estas modificaciones todas las frases del conjunto de entrenamiento pueden ser generadas y podemos proceder a calcular sus probabilidades de emisión:

$$P_{\theta}(\text{la vieja demanda ayuda})$$

$$= 1.0 * 0.4 * 1.0 * 0.1 * 0.3 * 0.3 * 0.2 + 1.0 * 0.3 * 1.0 * 0.3 * 0.2 * 0.7 * 0.2 = 0.00072 + 0.00252 = 0.00324$$

$$P_{\theta}(\text{la mujer oculta pelea})$$

$$= 1.0 * 0.4 * 1.0 * 0.3 * 0.3 * 0.1 * 0.2 + 1.0 * 0.2 * 1.0 * 0.3 * 0.7 * 0.7 * 0.4 = 0.00072 + 0.01176 = 0.01248$$

Como se puede observar en estos resultados, las probabilidades de emisión de estas frases cambian con respecto al ejemplo de las transparencias, puesto que la regla (Suj → Art Nom) ha cambiado su probabilidad inicial de 0.5 a 0.4 y ambas frases emplean dicha regla en una de sus derivaciones.

$$P_{\theta}(\text{la vieja mujer oculta demanda ayuda})$$

$$= 1.0 * 0.1 * 1.0 * 0.3 * 0.3 * 0.7 * 0.3 * 0.3 * 0.2 = 0.0001134$$

La tercera frase del conjunto de entrenamiento solo posee un árbol de derivación, el cual usa la nueva regla que hemos incluido. Con estas probabilidades de emisión procedemos a estimar la probabilidad de la regla ($Suj \rightarrow Art\ Nom\ Adj$) y del resto de reglas con la misma parte izquierda:

$$p(Suj \rightarrow Art\ Nom\ Adj) =$$

$$= \frac{\frac{1}{0.01248} \cdot 0.01176}{\frac{1}{0.00324} \cdot (0.00072 + 0.00252) + \frac{1}{0.01248} \cdot (0.00072 + 0.01176) + \frac{1}{0.0001134} \cdot 0.0001134}$$

$$= \frac{0.9289}{3} = 0.3141$$

$$p(Suj \rightarrow Art\ Nom) =$$

$$= \frac{\frac{1}{0.00324} \cdot 0.00072 + \frac{1}{0.01248} \cdot 0.00072}{\frac{1}{0.00324} \cdot (0.00072 + 0.00252) + \frac{1}{0.01248} \cdot (0.00072 + 0.01176) + \frac{1}{0.0001134} \cdot 0.0001134}$$

$$= \frac{0.2799}{3} = 0.0933$$

$$p(Suj \rightarrow Art\ Adj\ Nom) =$$

$$= \frac{\frac{1}{0.00324} \cdot 0.00252}{\frac{1}{0.00324} \cdot (0.00072 + 0.00252) + \frac{1}{0.01248} \cdot (0.00072 + 0.01176) + \frac{1}{0.0001134} \cdot 0.0001134}$$

$$= \frac{0.77}{3} = 0.259$$

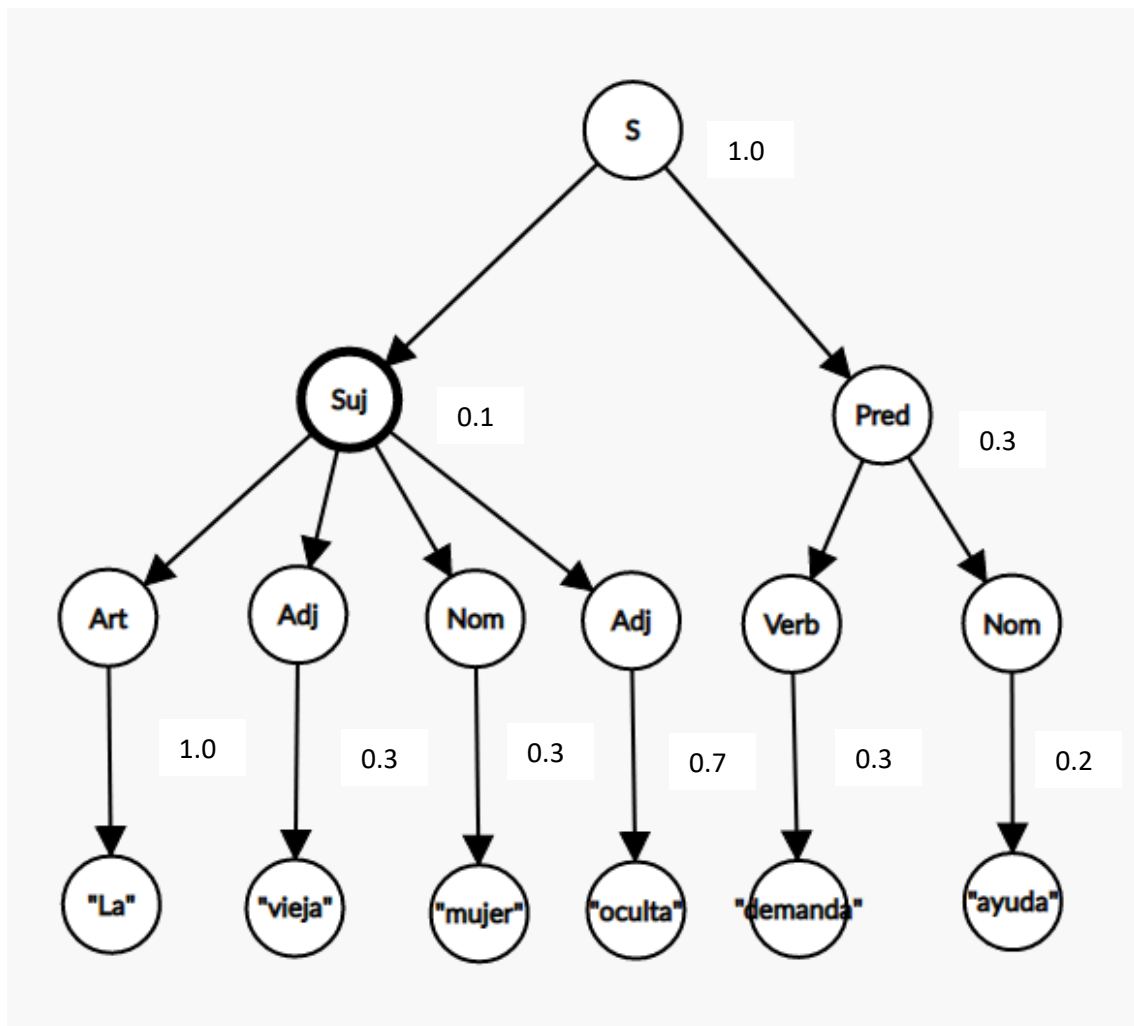
$$p(Suj \rightarrow Art\ Adj\ Nom\ Adj) =$$

$$= \frac{\frac{1}{0.0001134} \cdot 0.0001134}{\frac{1}{0.00324} \cdot (0.00072 + 0.00252) + \frac{1}{0.01248} \cdot (0.00072 + 0.01176) + \frac{1}{0.0001134} \cdot 0.0001134}$$

$$= \frac{1}{3} = 0.33$$

Como se puede ver la suma de las probabilidades estimadas de cada regla es igual a 1, por lo que los resultados son correctos.

La única derivación posible para la tercera frase del conjunto de entrenamiento tendrá el siguiente aspecto:



Question 6.

Si estimamos la probabilidad de la regla ($\text{Suj} \rightarrow \text{Art Nom Adj}$) a partir de las modificaciones realizadas sobre el ejemplo de las transparencias en el ejercicio anterior, el algoritmo k-best con $k=2$ otorgará el mismo resultado ya que ninguna de las frases de entrenamiento posee más de 2 árboles de derivación distintos, por lo que sería equivalente a aplicar el algoritmo Inside-Outside. Si queremos que existe una diferencia entre las estimaciones de ambos algoritmos podemos añadir más reglas de forma que alguna de las frases de entrenamiento llegue a tener 3 posibles arboles de derivación. Con este propósito se han vuelto a modificar las reglas y sus probabilidades iniciales de la siguiente forma:

$$p(\text{Suj} \rightarrow \text{Art Nom}) = 0.4$$

$$p(\text{Suj} \rightarrow \text{Art Adj Nom}) = 0.3$$

$$p(\text{Suj} \rightarrow \text{Art Nom Adj}) = 0.2$$

$$p(\text{Suj} \rightarrow \text{Art Adj Nom Adj}) = 0.1$$

$$p(\text{Pre} \rightarrow \text{Verb}) = 0.4$$

$$p(\text{Pre} \rightarrow \text{Verb Nom}) = 0.3$$

$$p(\text{Pre} \rightarrow \text{Adj Verb Nom}) = 0.2$$

$$p(\text{Pre} \rightarrow \text{Nom Adj Verb Nom}) = 0.1$$

Con estas nuevas reglas las frases “la vieja demanda ayuda” y “la mujer oculta pelea” mantienen sus posibles arboles de derivación:

$$P_{\theta}(\text{la vieja demanda ayuda})$$

$$= 1.0 * 0.4 * 1.0 * 0.1 * 0.3 * 0.3 * 0.2 + 1.0 * 0.3 * 1.0 * 0.3 * 0.2 * \\ * 0.4 * 0.2 = 0.00072 + 0.00144 = 0.00216$$

$$P_{\theta}(\text{la mujer oculta pelea})$$

$$= 1.0 * 0.4 * 1.0 * 0.3 * 0.3 * 0.1 * 0.2 + 1.0 * 0.2 * 1.0 * 0.3 * 0.7 * \\ * 0.4 * 0.4 = 0.00072 + 0.00672 = 0.00744$$

Pero la frase “la vieja mujer oculta demanda ayuda” pasa a tener 3 posibles árboles de derivación:

$$P_{\theta}(\text{la vieja mujer oculta demanda ayuda})$$

$$= 1.0 * 0.1 * 1.0 * 0.3 * 0.3 * 0.4 * 0.3 * 0.3 * 0.2 + 1.0 * 0.3 * 1.0 \\ * 0.3 * 0.3 * 0.2 * 0.7 * 0.3 * 0.2 + 1.0 * 0.4 * 1.0 * 0.1 * 0.1 * 0.3 * 0.7 \\ * 0.3 * 0.2 = 0.0000648 + 0.0002268 + 0.0000504 = 0.000342$$

Una vez calculadas las probabilidades de emisión procedemos a estimar la probabilidad de la regla ($\text{Suj} \rightarrow \text{Art Nom Adj}$) con el algoritmo Inside-Outside y el k-best.

Inside-Outside:

$$p(\text{Suj} \rightarrow \text{Art Nom Adj}) =$$

$$= \frac{\frac{1}{0.00744} \cdot 0.00672}{\frac{1}{0.00216} \cdot (0.00072 + 0.00144) + \frac{1}{0.00744} \cdot (0.00072 + 0.00672) + \frac{1}{0.000342} \cdot (6.48 * 10^{-5} + 2.268 * 10^{-4} + 5.04 * 10^{-5})} \\ = \frac{0.9032}{3} = 0.3010$$

K-best con k = 2:

$$p(\text{Suj} \rightarrow \text{Art Nom Adj}) =$$

$$\begin{aligned} &= \frac{\frac{1}{0.00744} \cdot 0.00672}{\frac{1}{0.00216} \cdot (0.00072 + 0.00144) + \frac{1}{0.00744} \cdot (0.00072 + 0.00672) + \frac{1}{0.0002916} \cdot (6.48 \cdot 10^{-5} + 2.268 \cdot 10^{-4})} \\ &= \frac{0.9032}{3} = 0.3010 \end{aligned}$$

Para el caso de la regla (Suj → Art Nom Adj) las formulas cambian pero el resultado continua siendo el mismo. Sin embargo si estimamos la regla (Suj → Art Nom) el resultado sí que varía ya que esta regla se emplea en la derivación menos probable de la frase “La vieja mujer oculta demanda ayuda”.

Inside-Outside:

$$p(\text{Suj} \rightarrow \text{Art Nom}) =$$

$$\begin{aligned} &= \frac{\frac{1}{0.00216} \cdot 0.00072 + \frac{1}{0.00744} \cdot 0.00072 + \frac{1}{0.000342} \cdot 5.04 \cdot 10^{-5}}{\frac{1}{0.00216} \cdot (0.00072 + 0.00144) + \frac{1}{0.00744} \cdot (0.00072 + 0.00672) + \frac{1}{0.000342} \cdot (6.48 \cdot 10^{-5} + 2.268 \cdot 10^{-4} + 5.04 \cdot 10^{-5})} \\ &= \frac{0.5775}{3} = 0.1924 \end{aligned}$$

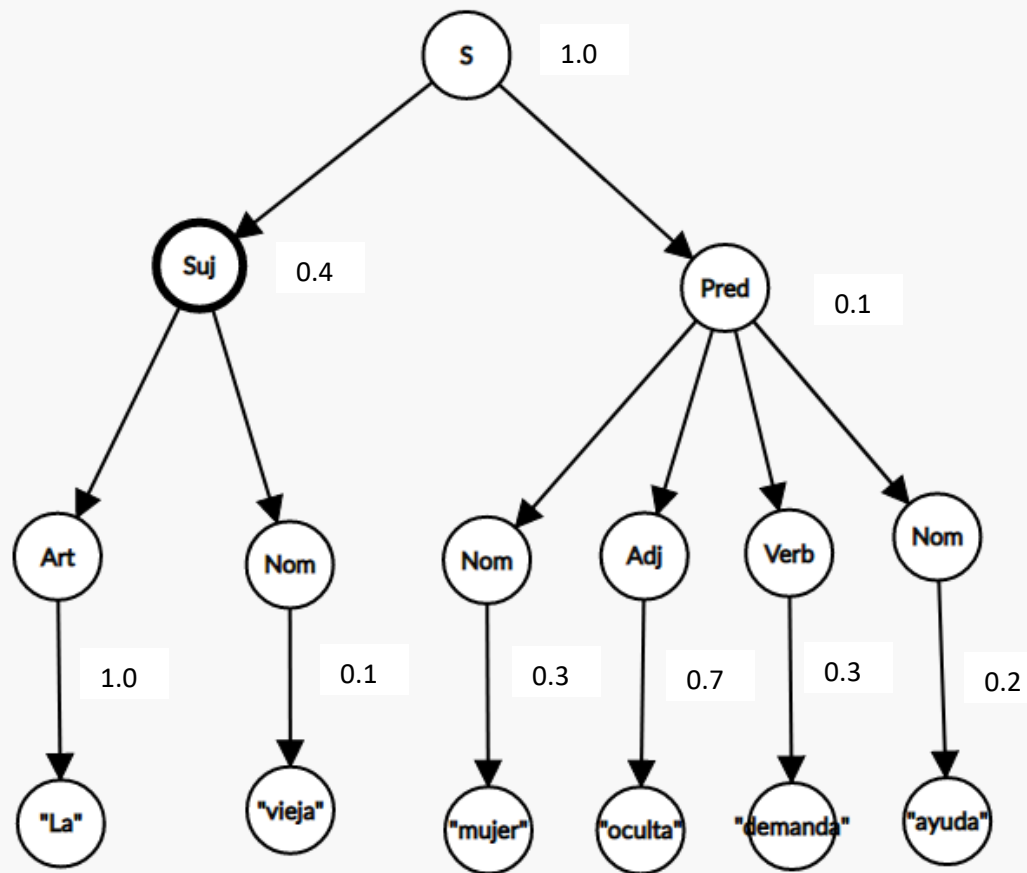
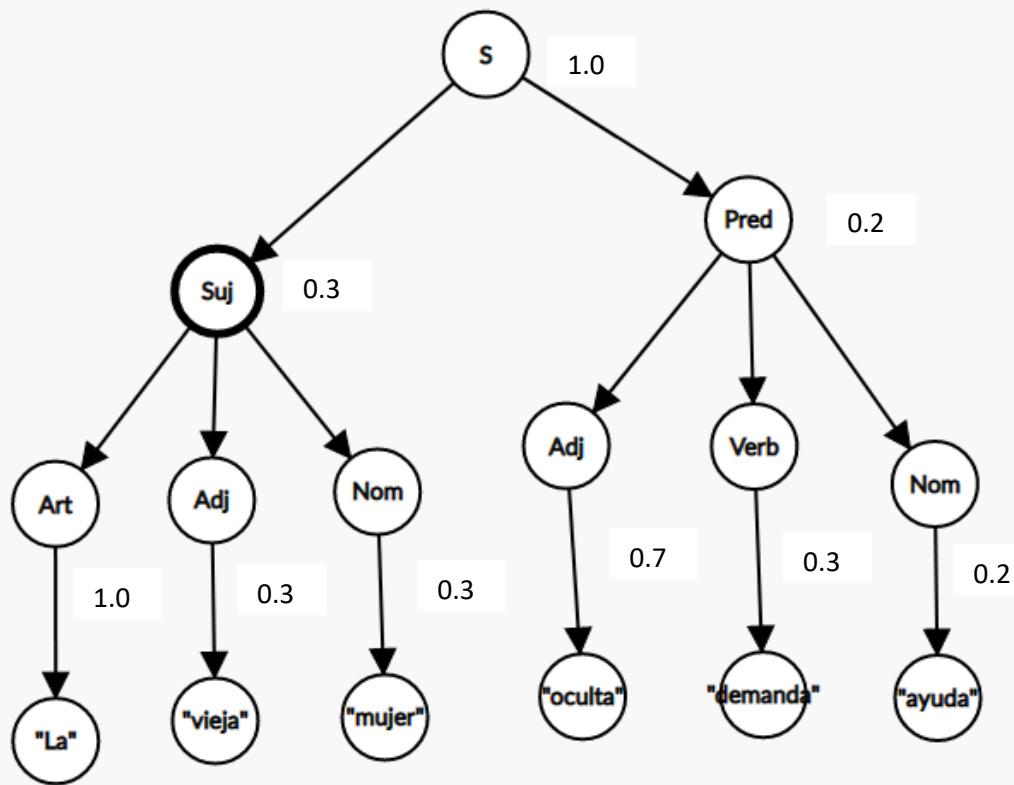
K-best con k = 2:

$$p(\text{Suj} \rightarrow \text{Art Nom}) =$$

$$\begin{aligned} &= \frac{\frac{1}{0.00216} \cdot 0.00072 + \frac{1}{0.00744} \cdot 0.00072}{\frac{1}{0.00216} \cdot (0.00072 + 0.00144) + \frac{1}{0.00744} \cdot (0.00072 + 0.00672) + \frac{1}{0.0002916} \cdot (6.48 \cdot 10^{-5} + 2.268 \cdot 10^{-4})} \\ &= \frac{0.4301}{3} = 0.1433 \end{aligned}$$

Como se puede observar en la estimación mediante k-best con k = 2, la estimación de la regla (Suj → Art Nom) obtiene un valor distinto.

Adicionalmente a la derivación del ejercicio anterior, las 2 nuevas derivaciones de la frase “la vieja mujer oculta demanda ayuda” tras la inserción de las nuevas reglas son:



Question 8.

El objetivo de este ejercicio es el de emplear el tool-kit SCFG para aprender 4 gramáticas a partir de un número determinada de no-terminales y un conjunto de muestras representando 10000 triángulos rectángulos, con el propósito de observar como el número de no-terminales disponibles en la gramática afecta a la precisión de esta.

Para lograr este objetivo hemos partido del modelo incluido con el tool-kit "G-triangle", el cual simplemente define 15 no-terminales y 3 terminales. Modificando el número de no-terminales definidos cuando sea necesario se han generado 4 nuevas gramáticas que aparecen en la siguiente tabla junto al número de triángulos rectángulos generados por dichas gramáticas:

Nº No-Terminales	Nº Triángulos Rectángulo
5	29
10	63
15	61
20	84

El aprendizaje de estas 4 gramáticas se ha realizado mediante la orden *scfg_learn* incluida en el tool-kit con parámetros necesarios para realizar 700 iteraciones y empleando como algoritmo de estimación Inside-Outside, a partir del conjunto de muestras *SampleTriangle-10K*.

Como se puede ver en los resultados de la tabla, a medida que aumentamos la cantidad de no-terminales disponibles para la gramática, más fácil le resulta aprender a generar triángulos rectángulos en un mismo número de iteraciones. Sin embargo, al mismo tiempo que aumenta la adaptación del modelo a las muestras también lo hace el tiempo de ejecución necesario para ello, ya que cada no-terminal genera nuevas posibles reglas que deben ser estimadas.

Question 9.

El objetivo de este ejercicio es el de analizar los resultados del aprendizaje de varios modelos centrados en la clasificación de triángulos equiláteros, isósceles y escalenos en base al algoritmo empleado. Al mismo tiempo también se compararán los resultados en función de si los modelos han sido entrenados sobre muestras de entrenamiento anotadas con corchetes o con muestras sin anotar.

Datos de entrenamiento con corchetes

Para lograr estos objetivos, en primer lugar partimos de los datos de entrenamiento disponibles con el Tool-kit SCFG que encontramos en los ficheros

“DATA/Tr-right”, “DATA/Tr-equil” y “DATA/Tr-isosc”, los cuales contienen, respectivamente, 1000 muestras anotadas con corchetes de triángulos escalenos, equiláteros e isósceles. A partir de estas muestras se han aprendido 3 modelos empleando el algoritmo Indide-Outside y otros 3 modelos empleando el algoritmo de Viterbi, de los cuales cada uno de los 3 modelos de cada tipo estará especializado en detectar un tipo de triángulo. Una vez obtenidos estos modelos, se ha realizado un cálculo de las probabilidades asignadas por cada modelo a las muestras de cada uno de los ficheros de test que encontramos en los ficheros “DATA/Ts-right”, “DATA/Ts-equil” y “DATA/Ts-isosc”; y se construido una matriz de confusión para representar la clasificación de dichas muestras en base a estas probabilidades.

```
raul@raul-VirtualBox:~/Desktop/
-toolkit/confus
```

	equi	isos	right	Err	Err%
equi	794	206	0	206	20.6
isos	531	225	244	775	77.5
right	108	145	747	253	25.3

Error: 1234/3000 = 41.13%

Ilustración 1: Matriz de confusión para muestras con corchetes y aprendidas con Inside-Outside

```
raul@raul-VirtualBox:~/Desktop/
oolkit/confus
```

	equi	isos	right	Err	Err%
equi	77	843	80	923	92.3
isos	70	850	80	150	15.0
right	12	676	312	688	68.8

Error: 1761/3000 = 58.70%

Ilustración 2: Matriz de confusión para muestras con corchetes y aprendidas con Viterbi

Tal y como se puede observar en los resultados, los resultados del algoritmo Inside-Outside proporcionan una menor tasa de error que los obtenidos por Viterbi, dado que durante la estimación de las reglas de los modelos, Inside-Outside emplea las probabilidades de emisión de todos los posibles árboles de derivación para cada muestra de entrenamiento, en contraste con Viterbi, que tan solo usa la probabilidad del árbol más probable.

A pesar de la mejora de resultados con respecto a Viterbi, el algoritmo Inside-Outside sigue generando un porcentaje de error considerable durante la clasificación, principalmente al clasificar las muestras de triángulos isósceles, las

cuales se clasifican erróneamente como equiláteros en más del 50% de las veces. En el resto de posibles categorías, el algoritmo Inside-Outside genera errores similares en torno al 20%, sin embargo en el caso de las muestras de triángulos equiláteros el total de este porcentaje se produce al calificar erróneamente las muestras como isósceles; mientras que en el caso de los escalenos el error se reparte entre las otras dos categorías.

Datos de entrenamiento sin corchetes

Para esta sección del ejercicio se ha procedido de forma idéntica a la interior, con la única diferencia de que previamente al aprendizaje de los modelos se han modificado los ficheros “DATA/Tr-right”, “DATA/Tr-equil” y “DATA/Tr-isosc” para eliminar los símbolos “[“ y “]”. De esta manera se consiguen unos conjuntos de entrenamiento cuyas muestras no limitan de forma tan restrictiva el número de posibles arboles de derivación para estas. Con este planteamiento se han alcanzado las siguientes matrices de confusión:

```
raul@raul-VirtualBox:~/Desktop/
lts | ./scfg-toolkit/confus
```

	equi	isos	right	Err	Err%
equi	783	217	0	217	21.7
isos	483	366	151	634	63.4
right	48	187	765	235	23.5

Error: 1086/3000 = 36.20%

Ilustración 3: Matriz de confusión para muestras sin corchetes y aprendidas con Inside-Outside

```
raul@raul-VirtualBox:~/Desktop/
lts-vi | ./scfg-toolkit/confus
```

	equi	isos	right	Err	Err%
equi	67	933	0	933	93.3
isos	171	612	217	388	38.8
right	55	372	573	427	42.7

Error: 1748/3000 = 58.27%

Ilustración 4: Matriz de confusión para muestras sin corchetes y aprendidas con Viterbi

De nuevo, al igual que sucedía con las muestras que sí incluían corchetes, el porcentaje de acierto de los modelos aprendidos por Inside-Outside resulta superior al de los modelos aprendidos por Viterbi. En este caso, para los modelos obtenidos con el algoritmo Viterbi ha mejorado el reconocimiento de triángulos escalenos y empeorado el de isósceles, aunque todavía existe una marcada tendencia a clasificar erróneamente los triángulos equiláteros como isósceles.

Por otra parte, el porcentaje de acierto de los modelos basados en Inside-Outside ha aumentado un 5% con respecto a los resultados de la sección anterior. La tendencia de clasificar triángulos isósceles como equiláteros se ha suavizado pero continúa existiendo.

Si bien los resultados obtenidos a partir de muestras sin corchetes presentan mejores resultados en cuanto al número de errores de clasificación cometidos, se debe tener en cuenta que también han resultado más costosos desde el punto de vista de los recursos de cómputo. En el caso del algoritmo Inside-Outside, el aprendizaje de un solo modelo a partir de muestras sin corchetes puede llegar a ocupar 8 horas ya que, tal y como se especifica en la documentación del Tool-kit SCFG, el aprendizaje de muestras sin corchetes tiene un coste temporal cúbico, mientras que para las muestras con corchetes este coste es lineal.

Question 10.

En este ejercicio se comparan los resultados de la clasificación de modelos obtenidos por Inside-Outside y Viterbi en base a la cantidad de muestras de entrenamiento disponibles. Para ello, en primer lugar se han generado 3 nuevos conjuntos de entrenamiento, cada compuesto por 10, 100 y 1000 muestras de cada uno de los 3 tipos posibles de triángulo (equilátero, isósceles y escaleno). A partir de estos 3 conjuntos de datos y con los conjuntos de test que acompañan al tool-kit, se ha procedido de manera similar al procedimiento del ejercicio 9, aprendiendo los modelos a partir de las muestras de entrenamiento con 100 iteraciones del algoritmo correspondiente, calculando las probabilidades de cada modelo y representando los resultados en una matriz de confusión. Los resultados para el algoritmo I-O han sido los siguientes:

```
raul@raul-VirtualBox:~/Desktop/Tool-kit SCFG/
o-10 | ./scfg-toolkit/confus
      equi isos righ Err Err%
equi  300  556  144  700 70.0
isos  224  487  289  513 51.3
righ   75  565  360  640 64.0
Error: 1853/3000 = 61.77%
```

Ilustración 7: I-O con 10 muestras

```
raul@raul-VirtualBox:~/Desktop/Tool-kit SCFG/
o-100 | ./scfg-toolkit/confus
      equi isos righ Err Err%
equi  377  295  328  623 62.3
isos  319  336  345  664 66.4
righ   93   99  808  192 19.2
Error: 1479/3000 = 49.30%
```

Ilustración 6: I-O con 100 muestras

```
raul@raul-VirtualBox:~/Desktop/Tool-kit SCFG/
o-1000 | ./scfg-toolkit/confus
      equi isos righ Err Err%
equi  634  289   77  366 36.6
isos  456  298  246  702 70.2
righ   90  142  768  232 23.2
Error: 1300/3000 = 43.33%
```

Ilustración 5: I-O con 1000 muestras

Tal y como se puede observar, la precisión de los modelos a la hora de clasificar las muestras de test aumenta junto con el número de muestras empleadas para el entrenamiento.

Resultados del algoritmo Viterbi:

```
raul@raul-VirtualBox:~/Desktop/PE
-10 | ./scfg-toolkit/confus
      equi isos righ  Err Err%
equi  794  206    0   206 20.6
isos  771  229    0   771 77.1
righ  622  177  201   799 79.9

Error: 1776/3000 = 59.20%
```

Ilustración 9: Viterbi con 10 muestras

```
raul@raul-VirtualBox:~/Desktop/PE
-100 | ./scfg-toolkit/confus
      equi isos righ  Err Err%
equi  218  782    0   782 78.2
isos  215  770   15   230 23.0
righ   26  595  379   621 62.1

Error: 1633/3000 = 54.43%
```

Ilustración 8: Viterbi con 100 muestras

```
raul@raul-VirtualBox:~/Desktop/PE
-1000 | ./scfg-toolkit/confus
      equi isos righ  Err Err%
equi  145  609  246   855 85.5
isos   98  748  154   252 25.2
righ    0  302  698   302 30.2

Error: 1409/3000 = 46.97%
```

Ilustración 10: Viterbi con 1000 muestras

Al igual que sucede con los resultados basados en I-O, la precisión de los modelos aprendidos por Viterbi aumenta junto con el número de muestras de entrenamiento. Sin embargo si comparamos los resultados de ambos algoritmos podemos ver que la precisión de Viterbi entrenado con 10 muestras es mayor que la precisión de I-O para el mismo número de muestras, mientras que para el resto las precisiones de I-O son siempre mejores. Esto podría sugerir que en caso de disponer de un bajo número de muestras de entrenamiento, el uso del algoritmo Viterbi puede ser preferible a I-O, o por lo menos una combinación de ambos algoritmos.