Instituto Politécnico Nacional
Escuela Superior de Cómputo
Unidad de Aprendizaje:

Procesamiento del Lenguaje Natural

Práctica 4:

N-gram model

Profesor: Joel Omar Juárez Gambino

Equipo 3:

- **Gutiérrez Pérez Gabriela G.**
- **Loeza Sebastián**
- **Ramos García Luis Gerardo**
- **Rico Mendoza Josué**

Fecha de entrega: 05 de noviembre 2024

Introducción:

Los modelos n-gram son modelos probabilísticos utilizados en procesamiento del lenguaje natural que analizan secuencias de palabras para predecir la probabilidad de aparición de una palabra en función de las palabras precedentes. Un n-grama puede ser un bigrama (dos palabras consecutivas), un trigrama (tres palabras), o más. Estos modelos asumen que cada palabra depende únicamente de las palabras anteriores en una ventana limitada, capturando el contexto inmediato.

Estos modelos usan probabilidades condicionales para calcular qué palabra es más probable que aparezca después de otra o de una secuencia específica, lo cual es útil para la generación de texto y predicción de palabras. Es importante mencionar que, para evitar probabilidades nulas a secuencias, generalmente se aplica un suavizado que da una probabilidad mínima a todos los n-gramas. La frecuencia de cada n-grama indica qué tan común es una secuencia particular de palabras, lo cual refleja tendencias lingüísticas y puede ayudar a mejorar la precisión de las predicciones. De manera que, las secuencias con mayor frecuencia reciben más peso en la generación de texto y en las tareas de predicción. En general, estos modelos son eficientes y de fácil implementación, aunque tienen limitaciones en su capacidad para capturar dependencias a largo plazo, también, los n-gramas capturan dependencias de contexto limitado (un término anterior en bigramas o dos en trigramas), lo cual permite entender el contexto inmediato, pero tiene limitaciones en secuencias más largas. Este modelo asume que la probabilidad de una palabra depende solo de las palabras anteriores en una secuencia limitada.

En esta práctica se desarrolló un modelo de lenguaje basado en bigramas y trigramas, el objetivo principal fue crear un modelo que analizara la secuencia y frecuencia de palabras, proporcionando una base para predicciones textuales y generación de oraciones.

La práctica se dividió en tres tareas clave: recopilación y tokenización de un corpus de mensajes por integrantes de equipo, creación de modelos de lenguaje para bigramas y trigramas, y generación de texto predictivo a partir del modelo creado. El proceso incluyó:

1. Cada miembro del equipo recopiló un conjunto de mensajes generados previamente, que luego fue procesado para crear un corpus individual. La tokenización dividió cada mensaje en palabras individuales generando versiones limpias y tokenizadas de cada corpus, facilitando el análisis de secuencias de palabras.
2. Con los corpus tokenizados, se generaron secuencias de palabras de dos (bigramas) y tres términos (trigramas). Esta segmentación permitió capturar dependencias contextuales en el texto, lo cual es esencial para predecir el próximo término con mayor precisión en función del contexto inmediato.
3. Se calculó la frecuencia de cada n-grama dentro del corpus. Esto incluyó tanto la frecuencia del n-grama en sí como la frecuencia del contexto que lo precede

(en el caso de los bigramas, es la primera palabra; para los trigramas, es el bigrama compuesto por las dos primeras palabras). Con estos datos, se calculó la probabilidad condicional de cada n-grama, lo que permite evaluar la probabilidad de que una palabra específica ocurra después de una o dos palabras dadas. La probabilidad condicional se calculó dividiendo la frecuencia del n-grama entre la frecuencia del contexto, proporcionando una base probabilística para predicciones.

Desarrollo:

1. Creación de modelos de lenguaje

Se realizo una ventana emergente el cual realiza lo siguiente:

1. Extracción y Filtrado de Mensajes

La primera parte del código se centra en extraer mensajes específicos de un archivo de texto que contiene registros de chats de WhatsApp.

- Extracción de Mensajes: Se abre el archivo y se leen todas las líneas.
- Filtrado: Cada línea es revisada para ver si cumple con ciertos requisitos:
    - Inicia con un formato de fecha y hora seguido de un nombre de quien envía el mensaje.
    - No contiene mensajes multimedia o eliminados.
    - No contiene enlaces web (http o www).
- Limpieza del Mensaje: Los mensajes que cumplen con los criterios se limpian para eliminar:
    - La fecha, hora y nombre del remitente.

    - Un aviso si el mensaje fue editado.
- Formato Final: Cada mensaje válido se guarda en una lista, rodeado de los símbolos $ al inicio y & al final.

2. Guardado de Mensajes en un CSV

La función para guardar en CSV toma la lista de mensajes limpios y los organiza en un archivo CSV. Cada mensaje se guarda en una fila bajo una columna llamada "Message".

3. Tokenización del Corpus

Para preparar los mensajes para el análisis de n-gramas (combinaciones de palabras), el siguiente paso es tokenizar el texto:

- Tokenización: Cada mensaje se divide en unidades mínimas (tokens), que pueden ser palabras o símbolos. Esto permite trabajar con fragmentos de texto separados.
- Conversión a Minúsculas: Las palabras se transforman a minúsculas para evitar confusiones entre formas mayúsculas y minúsculas.

4. Generación de N-gramas

El análisis se realiza sobre dos tipos de n-gramas: bigramas (combinaciones de dos palabras) y trigramas (combinaciones de tres palabras).

Para bigramas:

- Identificación de Bigramas: Se recorre el texto para extraer todas las secuencias de dos palabras consecutivas.
- Frecuencias y Probabilidades: Se cuenta la frecuencia de cada combinación y se calcula la probabilidad de cada bigrama en función del contexto. Esto indica la probabilidad de que una palabra siga a otra en el texto.
- Almacenamiento de Resultados: Cada bigrama y su probabilidad se guardan en un DataFrame (una estructura similar a una tabla) que luego se ordena por frecuencia.

Para trigramas:

- Identificación de Trigramas: Similar a los bigramas, pero esta vez se analizan secuencias de tres palabras.
- Frecuencias y Probabilidades: Se calcula cuántas veces aparece cada trigram y se determina la probabilidad en función de las dos palabras anteriores.
- Almacenamiento de Resultados: Los trigramas también se guardan en un DataFrame ordenado por frecuencia.

5. Guardado de los Resultados de N-gramas

Ambos DataFrames (bigramas y trigramas) se guardan en archivos CSV llamados bigrams.csv y trigrams.csv, para facilitar el análisis posterior.

6. Interfaz de Usuario

La interfaz gráfica permite al usuario interactuar con el programa de una forma sencilla:

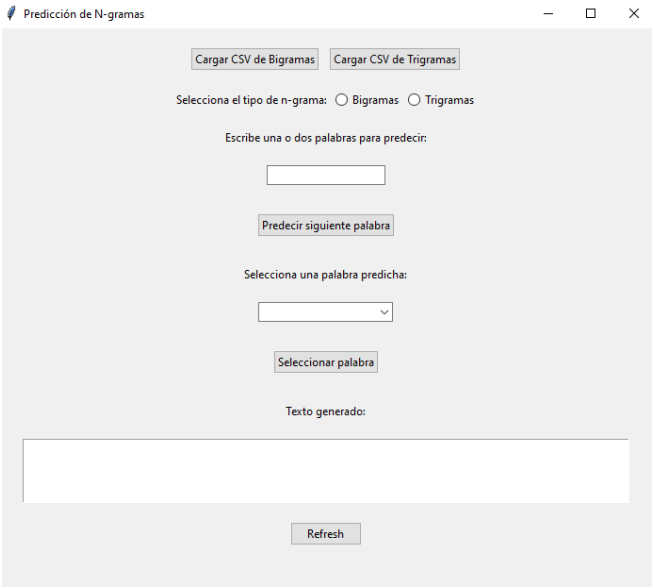
- Cargar Archivo CSV: Permite al usuario seleccionar el archivo que contiene los mensajes a analizar.
- Generar Bigramas y Trigramas: Una vez cargado el archivo, esta opción permite crear y guardar los archivos de bigramas y trigramas.
- Visualización en Pantalla: Los 10 bigramas y trigramas más comunes se muestran en la interfaz.

7. Mostrar Resultados en la Interfaz

Los resultados de los bigramas y trigramas se despliegan en una caja de texto en la interfaz, permitiendo al usuario ver las combinaciones de palabras más comunes sin necesidad de abrir los archivos CSV generados.

2. Ejercicio de texto predictivo

La funcionalidad de este ejercicio permite que, al cargar un archivo CSV que contienen bigramas o trigramas, el usuario ingrese una palabra y, al seleccionar el botón de 'predecir', el sistema devuelva las tres posibles continuaciones de la palabra junto con un carácter de finalización. De esta manera, se puede predecir la siguiente palabra.

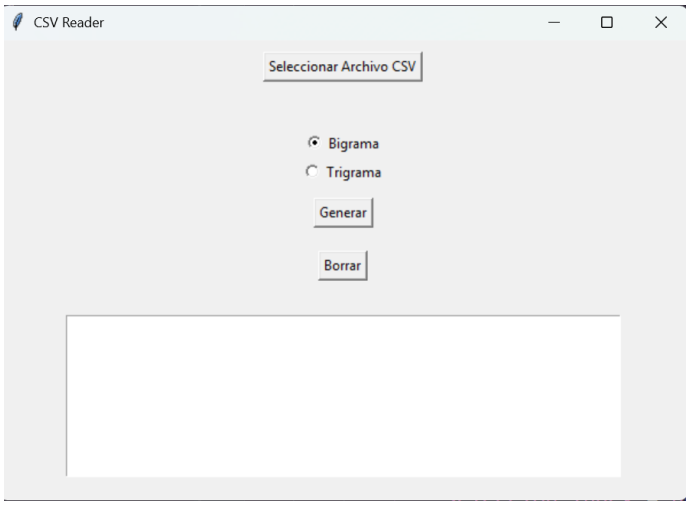


La interfaz desarrollada en Tkinter permite la predicción de texto utilizando un modelo probabilístico basado en bigramas o trigramas. El funcionamiento del programa se divide en varios pasos clave:

1. Cargar el modelo:
    - Se encuentran dos botones para cargar un único archivo CSV, ya sea de bigramas o trigramas.
    - Dependiendo del archivo cargado, hay un botón de opción (radio button) que indica con qué archivo se va a trabajar.
2. Escritura de palabra:
    - En el cuadro de texto, el usuario puede escribir una palabra para el modelo de bigramas o dos palabras para el modelo de trigramas.
3. Prediccion de la siguiente palaba:
    - Al hacer clic en el botón 'Predecir la siguiente palabra', se activa una función que predice las próximas palabras en un texto utilizando un modelo de lenguaje ya sea bigramas o trigrama. Esta función toma como entrada un modelo y una lista de una o dos palabras iniciales. Si se ingresa una sola palabra, busca las combinaciones más probables que comienzan con esa palabra. Si se ingresan dos palabras, busca las combinaciones que comienzan con ambas. Devuelve las n palabras más probables junto con sus respectivas probabilidades.
4. Selección de palabra:
    - Después de seleccionar la palabra, el texto inicial se trasladará al cuadro de texto que se encuentra al final de la ventana, y automáticamente esa palabra o las dos palabras (depende del modelo elegido) se colocará en el primer recuadro para repetir el mismo proceso. El resultado será un texto que se irá concatenando. En caso de querer finalizar, existe un carácter '.' que se puede utilizar para terminar el proceso.

3. Generación de texto

La interfaz desarrollada en Tkinter permite la generación de texto utilizando un modelo probabilístico basado en bigramas o trigramas. El funcionamiento del programa se divide en varios pasos clave:



1. Carga del modelo:
   - Al iniciar, el usuario carga un archivo CSV que contiene el modelo de texto de una persona particular. Este modelo puede ser de bigramas o trigramas.
   - Una vez cargado, el usuario selecciona el tipo de modelo que se usará (bigrama o trigrama) a través de un radio button, de forma que el programa sepa cómo estructurar las frases.
2. Generación de la frase:
   - Al presionar el botón "Generar", se inicia el proceso de creación de texto.
   - El primer paso es verificar que la cadena donde se almacenará la frase esté vacía. Si es así, el programa procede a buscar en el modelo probabilístico todas las palabras que suelen ser utilizadas para iniciar una oración.
   - Estas posibles palabras iniciales se introducen en una ruleta de selección, la cual asigna a cada palabra una probabilidad, basada en la frecuencia relativa en el modelo. Las probabilidades son normalizadas para que cada opción tenga su peso en la ruleta, y así se selecciona de manera aleatoria una palabra inicial para comenzar la frase.

3. **Construcción de la frase:**

   - A partir de la palabra inicial elegida, el programa continúa generando la frase:

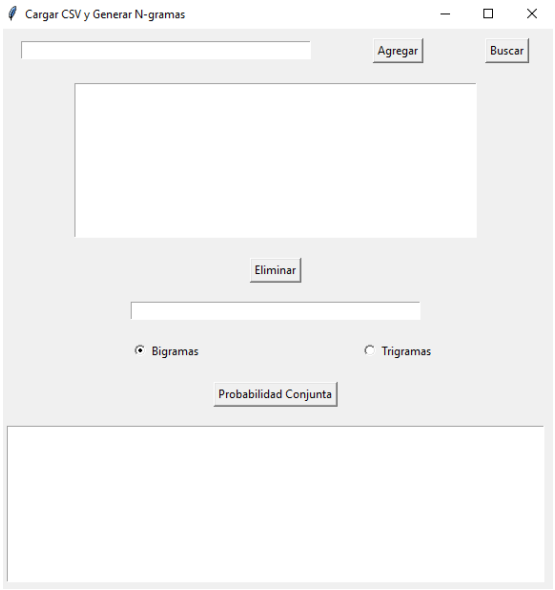
    - **Si el modelo es de bigramas**: el programa toma la última palabra agregada a la frase (contexto actual) y busca en el modelo probabilístico todas las posibles palabras que pueden seguirle.
    - **Si el modelo es de trigramas**: el programa utiliza las dos últimas palabras de la frase (contexto) y consulta el modelo para encontrar las posibles palabras que pueden venir a continuación.
- Una vez obtenidas las opciones de palabras que pueden seguir, se crea nuevamente una ruleta, asignando las probabilidades correspondientes a cada opción. Esta ruleta selecciona la palabra siguiente de forma aleatoria pero basada en la probabilidad asignada.

4. **Finalización de la frase:**
    - La palabra seleccionada se concatena a la frase, y el proceso se repite hasta que se obtenga una secuencia completa. En caso de que el modelo incluya un símbolo especial (como "&") que indica el fin de la oración, el programa detendrá la generación de texto.

5. **Función del botón limpiar:**
    - El botón "Limpiar" permite reiniciar el proceso, eliminando el contenido del área de texto y reiniciando la cadena de la frase a cero caracteres, lista para generar una nueva frase desde el inicio.

4. Probabilidad condicional

La funcionalidad de este ejercicio permite que, al cargar archivos CSV que contienen bigramas o trigramas, el usuario pueda ingresar cualquier frase y obtener la probabilidad de que dicha frase aparezca en cada uno de los modelos cargados.



El funcionamiento del programa se divide en varios pasos clave:

1. Carga del modelo:
    - Se creó una ventana utilizando la libreria Tkinter, en la cual atraves del cuadro de texto inicial se puede ingresar la ruta de los archivos CSV (ya sean de bigramas o trigramas), de otro modo, el botón "Buscar" facilita la selección del archivo al abrir el explorador de archivos de Windows.
    - Una vez seleccionado el archivo, se puede hacer clic en "Agregar" para cargarlo en el sistema.
    - En caso de haber seleccionado el archivo incorrecto, es posible eliminarlo del sistema seleccionándolo y haciendo clic en el botón "Eliminar", se pueden cargar N archivos csv.
2. Probabilidades conjuntas
    - En el segundo recuadro de texto, el usuario puede escribir cualquier frase y seleccionar si se trabajará con bigramas o trigramas
    - La frase se convierte en un archivo CSV que se estructura en bigramas o trigramas, según la selección del usuario.
    - Para cada bigrama o trigrama de la frase, se busca su correspondencia en el primer archivo CSV cargado. Si se encuentra, se recupera la probabilidad asociada.
    - El proceso continúa con el siguiente bigrama o trigrama, y si también se encuentra en el archivo, las probabilidades se multiplican. En caso de no encontrarlo, se multiplica por uno.

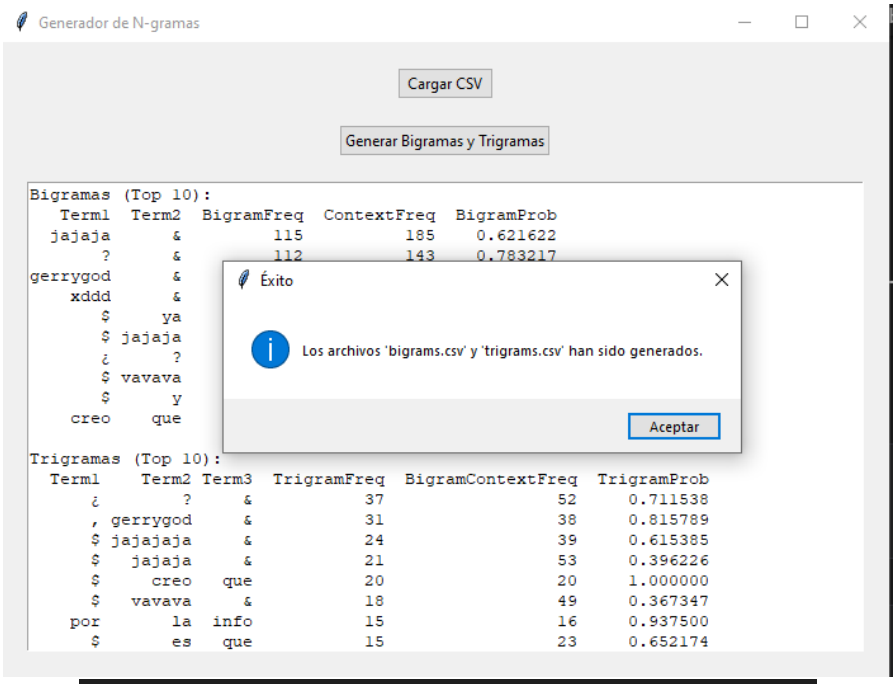
- Este proceso se repite hasta que se procesan todos los bigramas o trigramas de la frase y obtener la multiplicación de todas probabilidades, luego se realiza de nuevo con cada archivo CSV cargado.

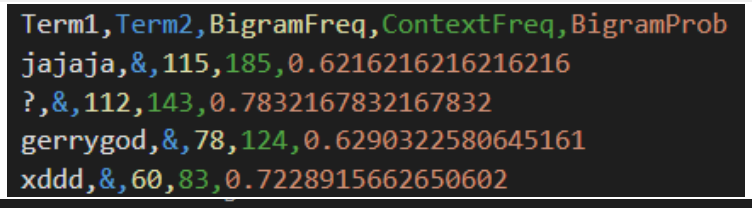
3. Resultados
    - Finalmente, se muestran los resultados, donde se indica la probabilidad de que la frase ingresada se encuentre en cada uno de los archivos CSV.

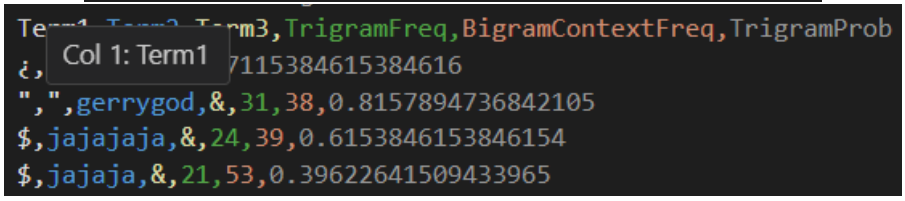
Experimentos:

Creación de modelos de lenguaje

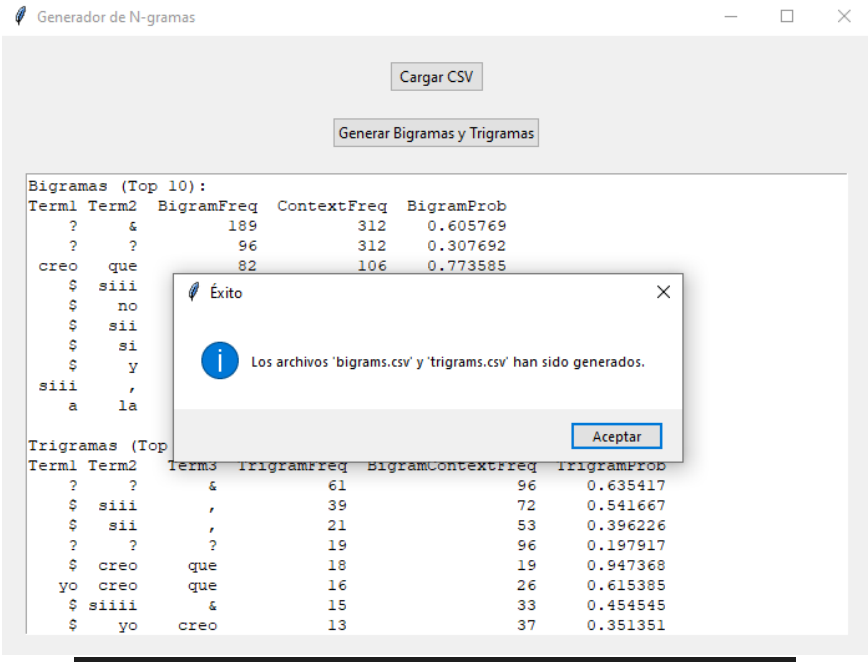
- Ejemplo 1 (compañero 1)







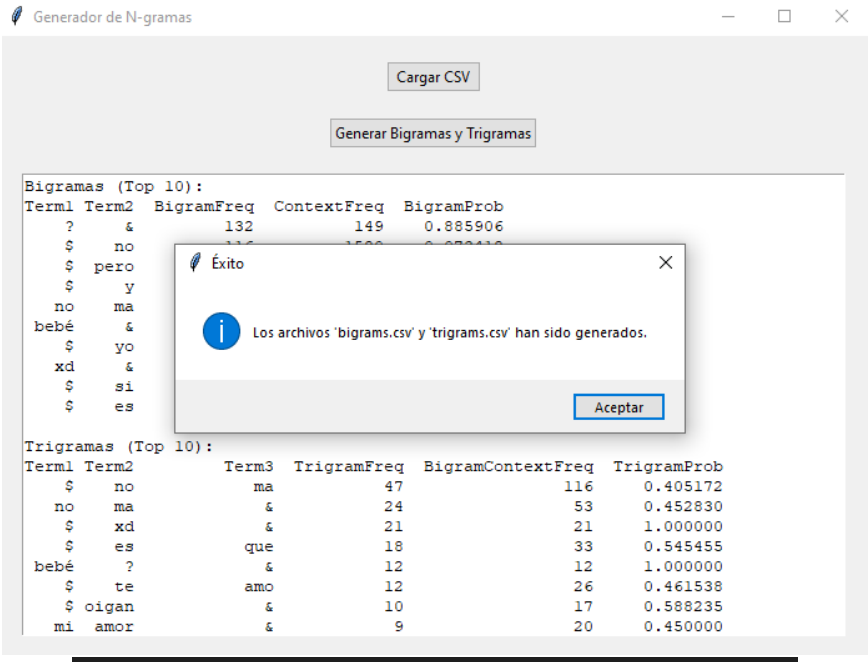
- Ejemplo 2 (compañero 2)




```
Term1,Term2,BigramFreq,ContextFreq,BigramProb
?,&,189,312,0.6057692307692307
?,?,96,312,0.3076923076923077
creo,que,82,106,0.7735849056603774
$,siii,72,1514,0.047556142668428
```

```
Term1,Term2,Term3,TrigramFreq,BigramContextFreq,TrigramProb
?,?,&,61,96,0.6354166666666666
$,siii,",",39,72,0.5416666666666666
$,sii,",",21,53,0.39622641509433965
?,?,?,19,96,0.19791666666666666
```

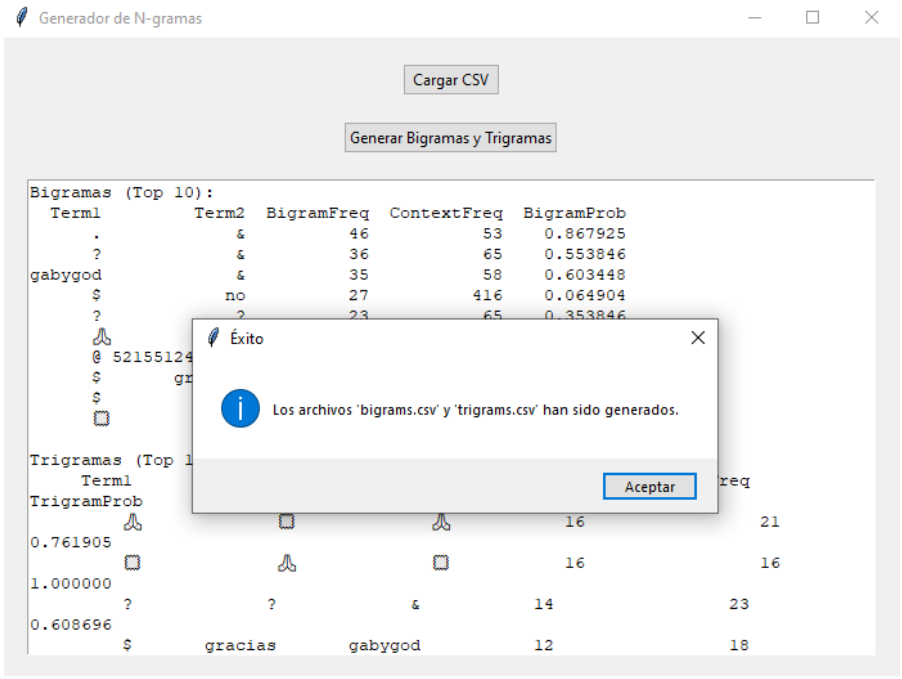
- Ejemplo 3 (compañero 3)




```
Term1,Term2,BigramFreq,ContextFreq,BigramProb
?,&,132,149,0.8859060402684564
$,no,116,1580,0.07341772151898734
$,pero,64,1580,0.04050632911392405
$,y,64,1580,0.04050632911392405
```

```
Term1,Term2,Term3,TrigramFreq,BigramContextFreq,TrigramProb
$,no,ma,47,116,0.4051724137931034
no,ma,&,24,53,0.4528301886792453
$,xd,&,21,21,1.0
$,es,que,18,33,0.5454545454545454
```

- Ejemplo 4 (compañero 4)

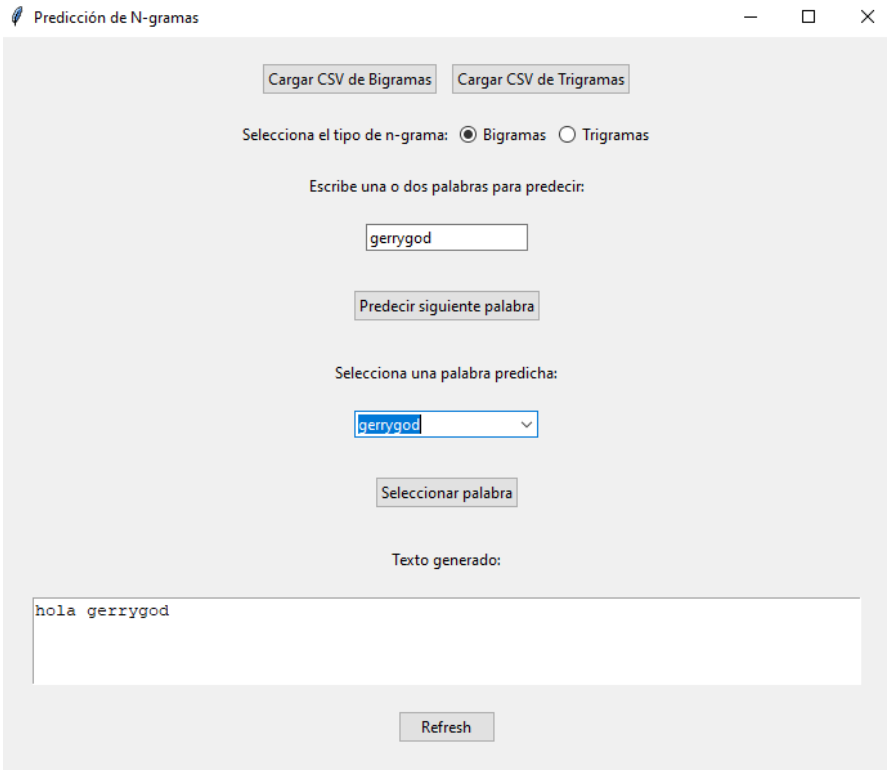



```
Term1,Term2,BigramFreq,ContextFreq,BigramProb
por,que,4,20,0.2
va,a,4,8,0.5
con,el,4,18,0.2222222222222222
que,nos,4,61,0.06557377049180328
sagal,&,4,9,0.4444444444444444
no,te,4,56,0.07142857142857142
```

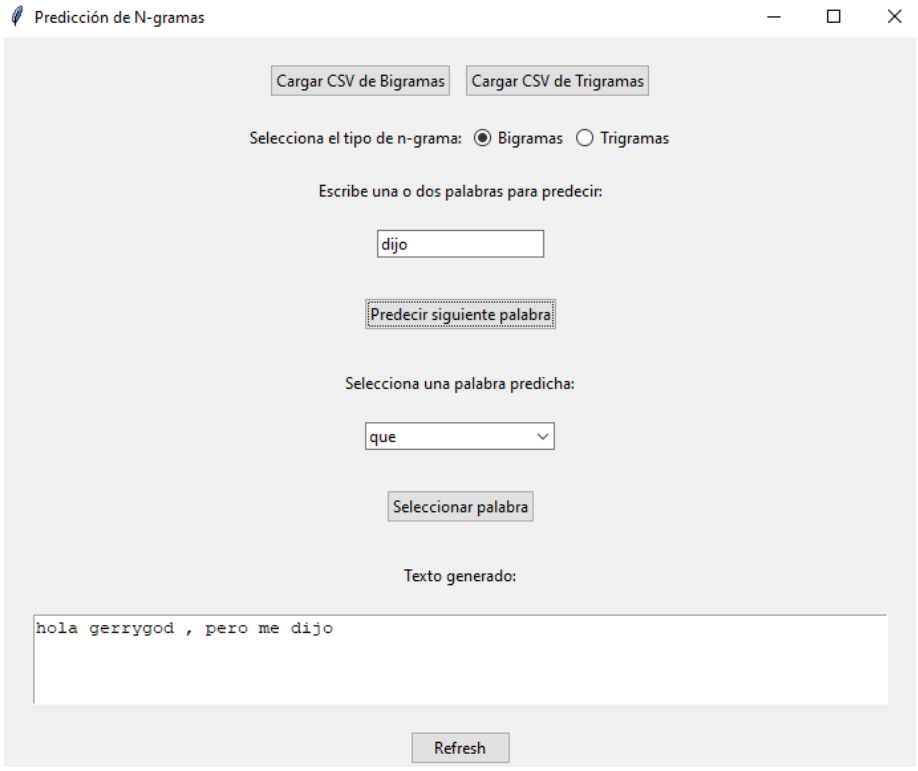
```
Term1,Term2,Term3,TrigramFreq,BigramContextFreq,TrigramProb
wey,?,&,2,2,1.0
el,protocolo,&,2,6,0.3333333333333333
$,si,&,2,11,0.18181818181818182
$,y,al,2,10,0.2
$,lo,que,2,6,0.3333333333333333
```

Ejercicio de texto predictivo

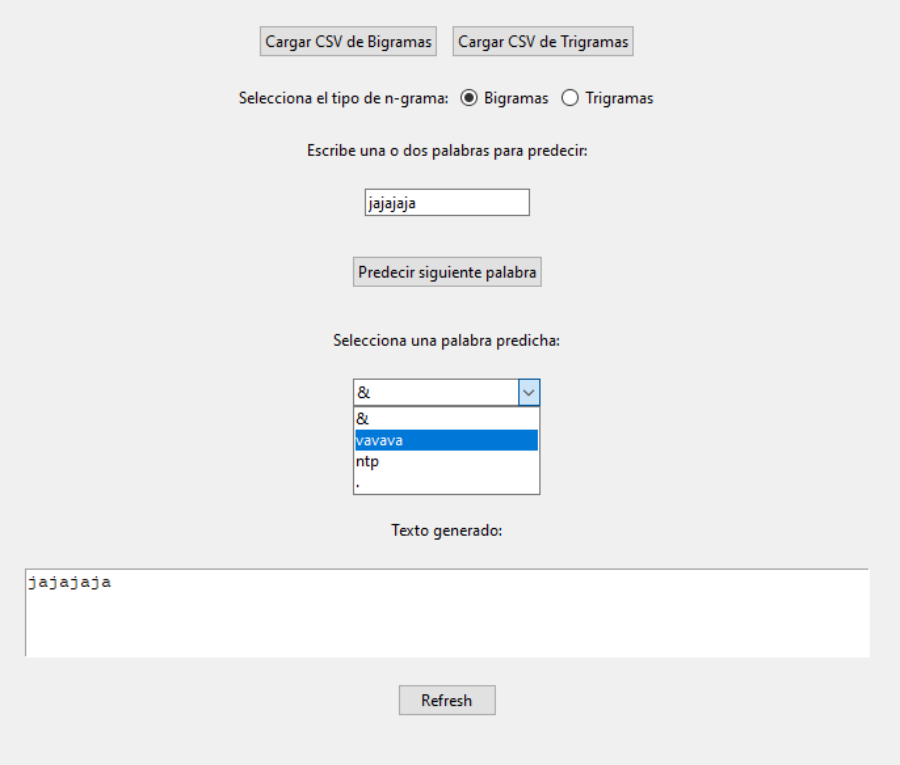
- Ejemplo 1 (Bigrama)



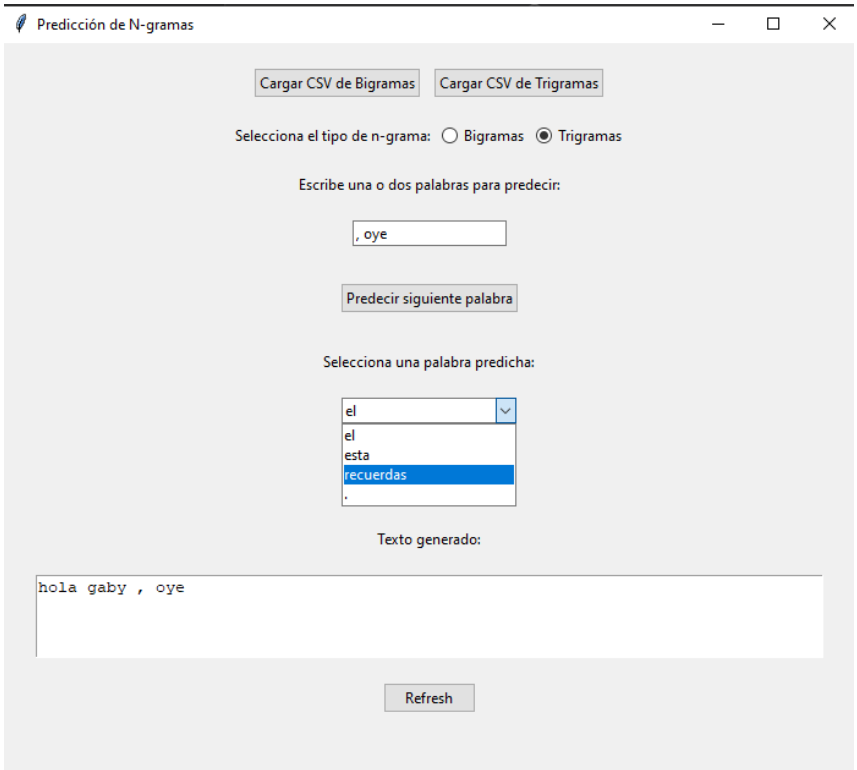
- Ejemplo 2 (Bigrama)



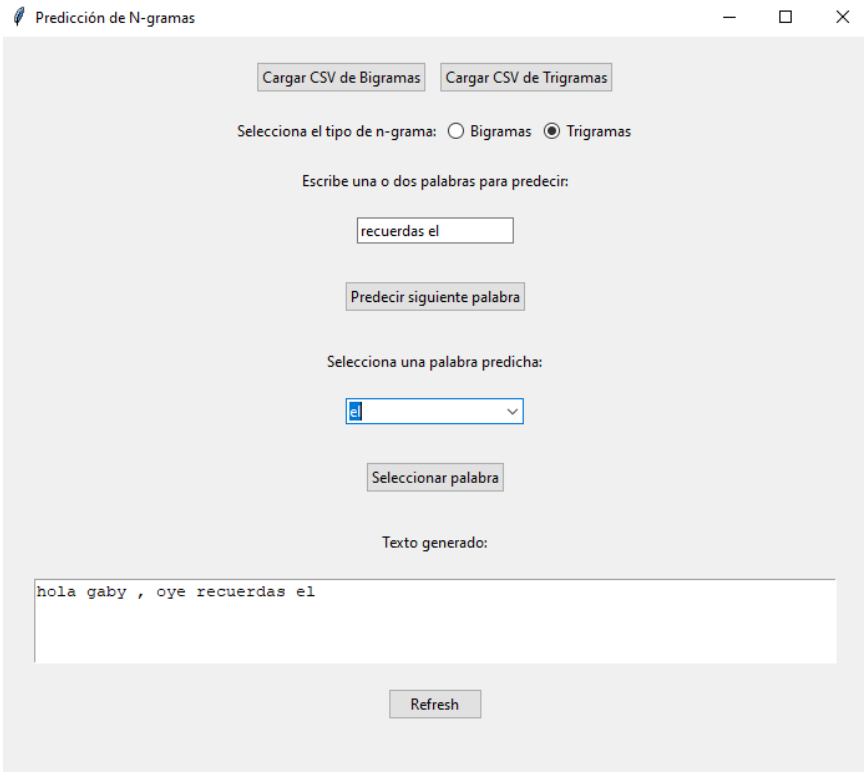
- Ejemplo 3 (Bigrama)



- Ejemplo 4 (Trigrama)

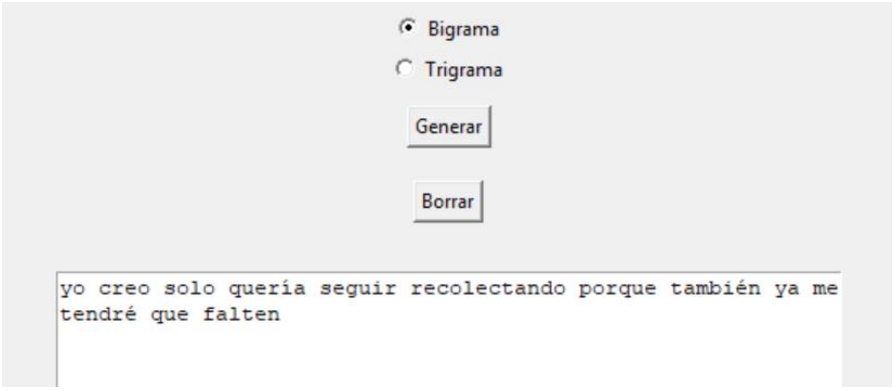


- Ejemplo 5 (Trigrama)

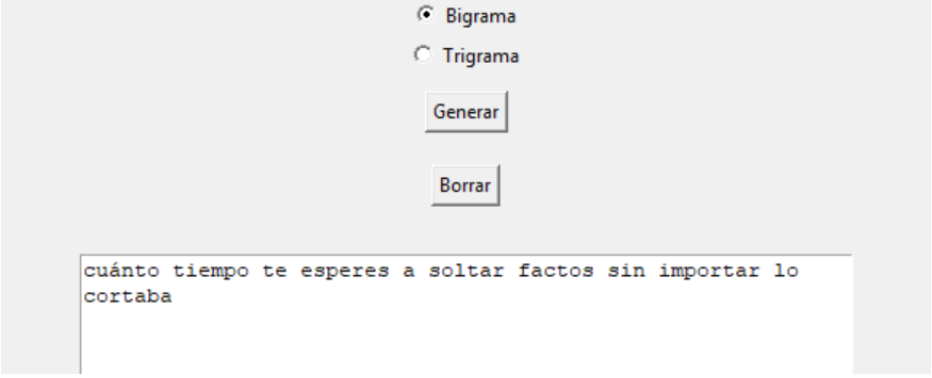


Ejercicio de generación de texto:

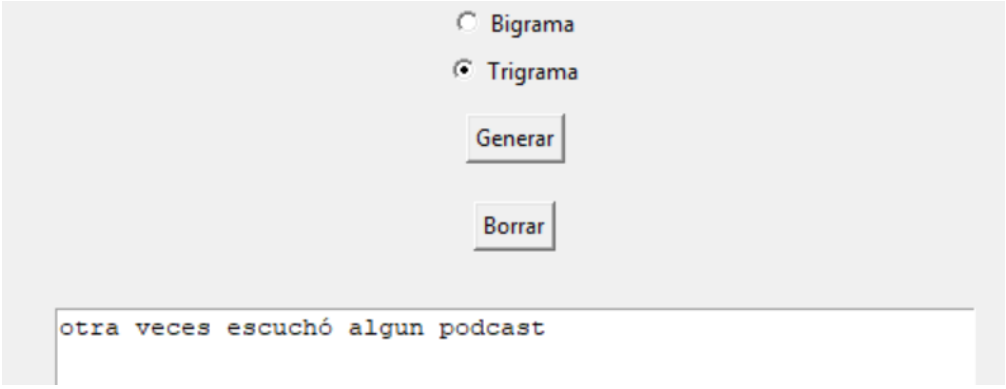
- Ejemplo 1 (Bigrama)



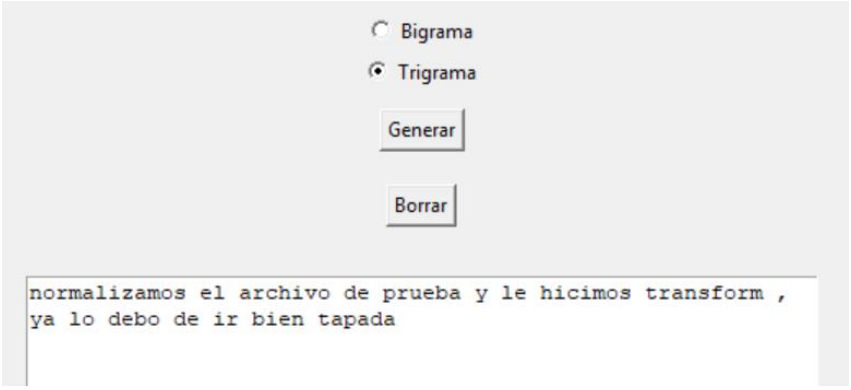
- Ejemplo 2 (Bigrama)



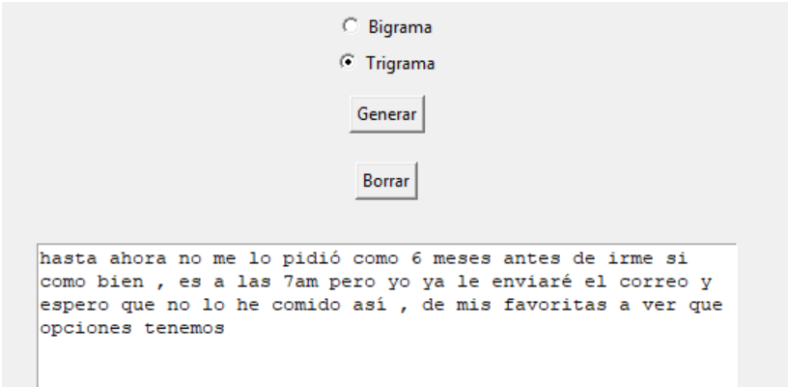
- Ejemplo 3 (Trigrama)



- Ejemplo 4 (Trigrama)

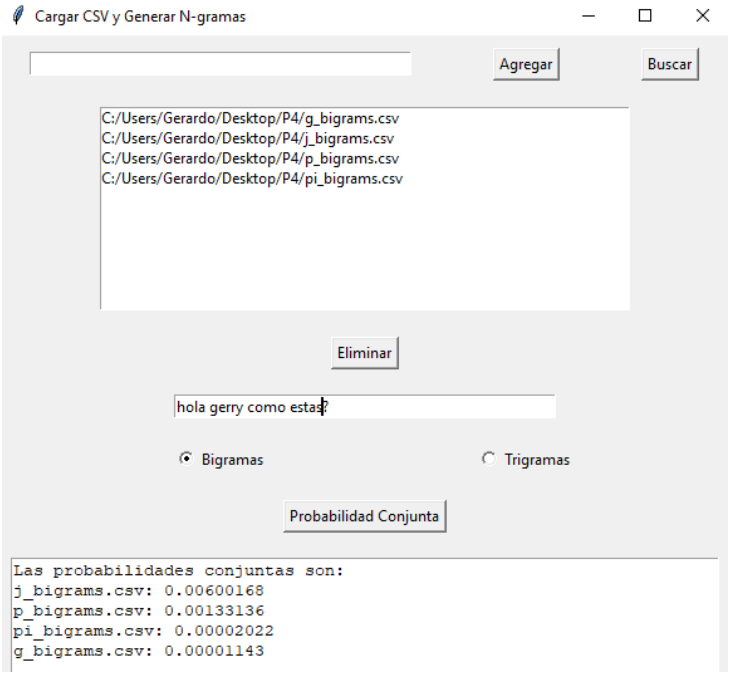


- Ejemplo 5 (Trigrama)

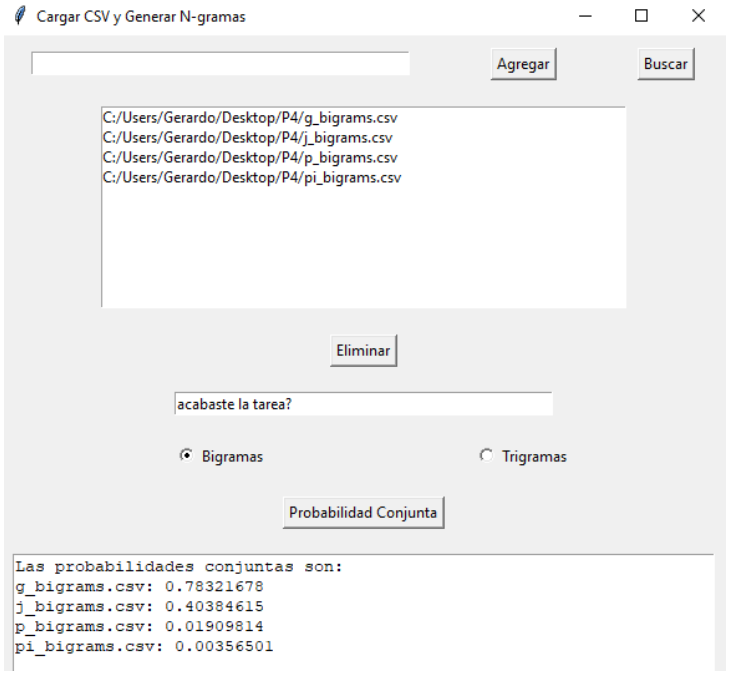


Ejercicio de probabilidad condicional

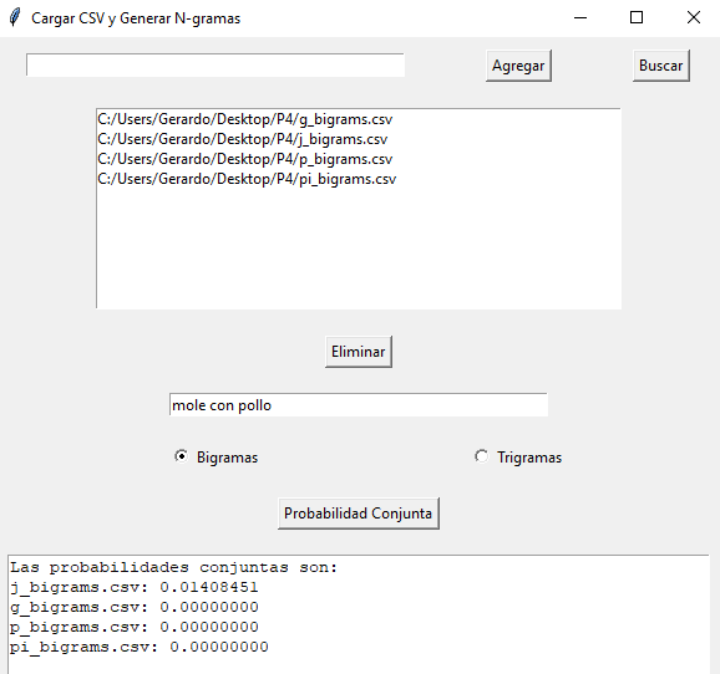
- Ejemplo 1 (Bigrama)



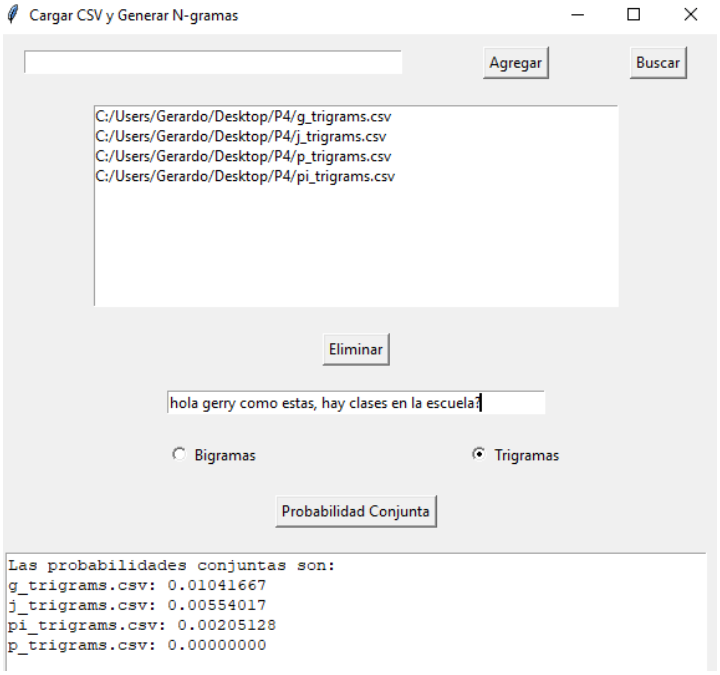
- Ejemplo 2 (Bigrama)



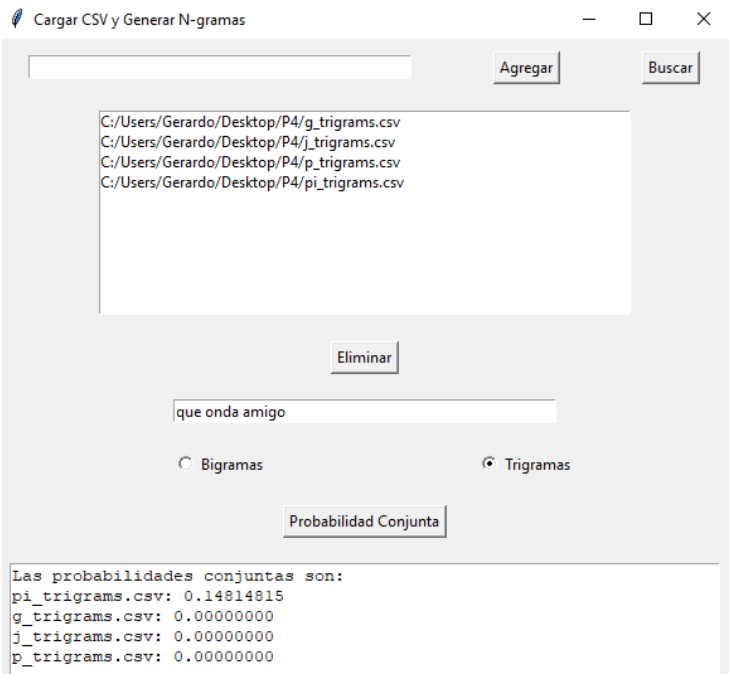
- Ejemplo 3 (Bigrama)



- Ejemplo 4 (Trigrama)



- Ejemplo 5 (Trigrama)



Conclusiones

Los n-gramas ofrecen una base sólida para la creación de modelos de lenguaje debido a su capacidad para capturar las secuencias de palabras más probables en un texto. Al calcular la probabilidad conjunta de que una secuencia de palabras ocurra, podemos predecir con mayor precisión la siguiente palabra en una oración. Asimismo, para evitar problemas con palabras o secuencias poco frecuentes, se emplean técnicas de suavizado que asignan una probabilidad mínima a todos los posibles n-gramas, de manera que, la calidad de los modelos n-grama depende en gran medida del tamaño y calidad del corpus, así como de las técnicas de suavizado empleadas. Esta combinación de probabilidad conjunta y suavizado permite crear modelos de lenguaje robustos y eficientes, que encuentran aplicación en diversas tareas como la generación de texto, la traducción automática, el reconocimiento de voz y la búsqueda de información.

Los n-gramas, al ser relativamente simples de implementar y entender, son un punto de partida ideal para explorar modelos de lenguaje más complejos y sofisticados.