503202/503203 Programación
Tarea Semestral

EQUIPO PROGRAMACIÓN

11 de abril de 2024

Introducción

El propósito de esta tarea es familiarizarse con el proceso de análisis, diseño e implementación de la solución a un problema de programación de mediana a alta complejidad usando lenguaje Python. También se espera que a través de la experiencia en este trabajo se potencien las habilidades de trabajo en equipo.

El trabajo debe ser realizado en grupos de 3 ó 4 estudiantes, quedando absolutamente prohibido el trabajo individual o en una configuración distinta de equipos. Los equipos se deben formar con estudiantes de una misma sección, no está permitido que en un equipo se mezclen estudiantes de secciones distintas.

Este enunciado se entregará a través de la plataforma CANVAS y en el caso de requerir aclaración de alguno de los aspectos expuestos en el documento, se recomienda que la comunicación se haga electrónicamente, de preferencia usando CANVAS, para hacer extensivas las explicaciones al resto del curso.

PROBLEMA: Predicción de Texto

Algunos editores de texto como Word o los clientes de correo electrónico como Gmail suelen incorporar "predictores de texto" que tienen como función *adivinar* la o las siguientes palabras que uno escribirá en base a la última palabra efectivamente escrita. Una de las estrategias más simples para hacer esto consiste en que, dada una palabra semilla (la última palabra escrita), el programa predictor analiza un "corpus" de palabras, por ejemplo, las palabras escritas en un libro o documento, en el cual se busca la palabra semilla y todas las palabras que siguen a esta en el "corpus", calculando la frecuencia de cada una y con ello la probabilidad de ocurrencia de cada una después de la palabra semilla.

A modo de ejemplo, consideremos que la palabra semilla ingresada al programa es "**mi**" y el corpus es el texto "**Ayer saqué a pasear a mi perro. Al salir mi perro mordió al perro de mi vecino**.". En este caso el predictor tiene dos palabras que suceden a **mi** en el corpus: {**perro**,**vecino**}, la primera aparece dos veces después de **mi** y la segunda sólo una vez, por tanto, las probabilidades de ocurrencia de cada una son $\frac{2}{3}$ y $\frac{1}{3}$, respectivamente. Así, en el ejemplo, el, predictor completaría la frase con "**mi** perro".

Para esta tarea se pide implementar un **programa predictor de texto** basado en la lógica anteriormente descrita, el cual tendrá que *adivinar* N palabras p_1, p_2, ..., p_N de un texto, usando una palabra semilla p_0, donde las palabras p_i se deducirán de un corpus a partir de la palabra p_{i-1}, con $i = 1, 2, ..., N$, es decir, p_1 se deduce de p_0, p_2 se deduce de p_1, p_3 se deduce de p_2, y así sucesivamente.

Existe la posibilidad de que un conjunto de m palabras W_1, W_2,.., W_m suceda a otra palabra X en el corpus con la misma probabilidad, siendo esta la probabilidad mayor. En este caso, el programa debe elegir al azar una de las palabras W_j, con $1 \leq j \leq m$.

La elección del *corpus* es libre, teniendo en cuenta que el programa funcionará mejor si se entrena con un corpus más extenso. Usted puede probar con diferentes textos a su elección. Al momento de evaluar el programa, se probará todas las tareas con un corpus específico, con el cual la predicción de texto con palabra siguiente más frecuente debería tener la misma respuesta.

Entradas:

Este programa tendrá tres entradas, el corpus o texto libre, compuesto de palabras separadas por uno o más espacios y signos de puntuación (coma, punto, punto y coma, etc.) propios del lenguaje escrito español. Las palabras estarán compuestas por letras mayúsculas y/o minúsculas, incluyendo vocales con tilde y las letras ñ y Ñ. No habrá dígitos ni números en el corpus. La segunda entrada corresponderá a sólo una palabra, compuesta de una o más letras (mayúsculas, minúsculas, con o sin tilde), la cual tendrá como requisito que debe pertenecer al corpus, si la palabra está mal ingresada o no pertenece al corpus debe reingresar otra palabra hasta que la entrada sea correcta. Por último, la tercera entrada será un número entero positivo N, cuyo valor no puede superar el 10 % del tamaño del corpus, en cantidad de palabras (es decir, si el corpus tiene 100 palabras el valor de N debe estar entre 1 y 10). Si el valor de N está mal ingresado se debe reingresar hasta que su valor sea correcto.

Salidas:

La única salida del programa debe ser una frase de $N + 1$ palabras, la primera la semilla p_0 y las restantes N las predichas p_1 p_2 ... p_N.

Ejemplo de entrada:

```
Ayer saqué a pasear a mi perro. Al salir, mi perro mordió al perro de mi vecino. Yo no entiendo
que sucedió con mi perro, él siempre es muy dócil.
mi
3
```

Ejemplo de salida:

En este caso la salida podría ser una de las siguientes frases:

```
mi perro al
mi perro mordió
mi perro de
mi perro él
```

Observación:

En el caso del ejemplo, *perro* tiene una probabilidad de 0.75 después de la palabra *mi* (en contraste con el 0.25 de probabilidad de la palabra *vecino*). Luego, todas las palabras *al*, *mordió*, *de* y *él*, tienen probabilidad de 0.25 de aparecer después de *perro*, por tanto, se elige al azar.

Instrucciones

Este trabajo debe ser realizado por los equipos establecidos en CANVAS, cuyos integrantes deben designar un/a **coordinador/a** o **líder**, quien será el/la que se comunique con el/la profesor/a de su sección o coordinador en caso de requerirlo.

El plazo máximo para inscribir los equipos vence el 26 de abril de 2024.

Los/as alumnos/as que no estén registrados en ningún equipo serán designados en equipos conformados por el coordinador del curso.

La única entrega de la tarea será, como máximo, el 14 de junio de 2024, para lo cual cada equipo debe preparar lo siguiente:

1. un archivo denominado `INFORME.PDF` que contenga:

- una **portada** en que figuren, como mínimo, los nombres de los/as integrantes del equipo, carrera de cada uno/a y el nombre del curso (sección).
- En su interior, el informe debe establecer qué funciones fueron asignadas a cada miembro del grupo de trabajo, cuántas reuniones realizaron y cuál fue el objetivo de cada una, indicando además quiénes participaron en cada reunión, es decir, la **organización del equipo**,
- se debe además describir el **desarrollo** indicando la forma en que se realizó el trabajo (**análisis del problema** (aclaración de los aspectos del enunciado que no fueron abordados o están poco claros, indicando para cada aspecto las opciones revisadas y las opción elegida y su justificación), **estrategias de solución** (describa detalladamente el método matemático, geométrico, estadístico u otra materia que permite resolver el problema), **implementación de la solución** (describa los algoritmos, las funciones, las variables y las estructuras de datos usadas y que implementan la solución al problema), incluya algunas **pruebas** realizadas con sus datos.
- Finalmente, el informe debe incluir también algunas **conclusiones generales** del trabajo.

No incluya código en el informe.

2. un archivo denominado `readme.txt` que contenga los nombres de los integrantes del equipo y las instrucciones, tanto para ejecutar su programa. Las indicaciones en este archivo se seguirán estrictamente, por lo tanto, es su responsabilidad ser claro en el orden y especificidad de las instrucciones.

3. uno o más archivos `*.py` o `*.ipynb` con el código en lenguaje Python . Los programas deben incluir (en líneas de comentarios) los nombres completos de los integrantes del grupo además de comentarios que permitan identificar las partes principales de los programas.

Todos los archivos mencionados deben comprimirse en un paquete ZIP o RAR con el nombre identificatorio de su equipo, por ejemplo "Equipo_001.ZIP", el cual debe ser enviado a través de la plataforma CANVAS.

No se aceptarán tareas incompletas, esto es, la tarea consiste en el programa funcionando y el informe. Si falta alguna de estas secciones el resultado será una nota reprobatoria.

- Asegurarse de escribir la información de identificación del grupo en todos los archivos.
- Eliminar despliegues innecesarios en el(los) programa(s).
- No serán recibidas partes o la totalidad de la tarea en forma impresa.
- No envíe archivos ejecutables (*.EXE), la plataforma y algunos clientes de correo electrónico no lo permiten.
- Si después de enviar la tarea descubre un error y se encuentra dentro del plazo de entrega, el grupo puede enviar una copia corregida, para ello hacer otro paquete ZIP o RAR, y volver a depositar (nos quedaremos con la última copia enviada).

Evaluación

El código será compilado, ejecutado y probado en una máquina usando Windows y Jupyter Notebook, por lo tanto, si algún grupo requiere la instalación de alguna biblioteca especial o aplicación complementaria, o bien, si es necesario compilar o correr la aplicación en un sistema operativo diferente, debe informarlo al profesor usando el archivo `readme.txt`.

El puntaje máximo de la tarea es 60 puntos distribuidos de la siguiente forma:

1 Entrega oportuna.

3 Cumple con especificaciones de entrega del trabajo incluyendo la estructura definida para el informe.

3 Describe adecuadamente la organización del equipo.

10 Realiza un análisis del enunciado y datos con el fin de diseñar una estrategia de solución al problema.

10 Describe las estrategias de solución al problema.

10 Describe los algoritmos, las funciones, las estructuras de datos y realiza las pruebas solicitadas.

3 Expone conclusiones adecuadas al desarrollo del trabajo.

20 Correctitud del código cuando es ejecutado. Se realizará una prueba exhaustiva de todas las funciones solicitadas, incluyendo ingreso de datos fuera de rango, para los cuales el programa debe responder correctamente.

El informe debe estar escrito correctamente, bien diagramado y sin problemas de ortografía y redacción (se descontará 2 décimas de su nota por cada falta de ortografía en el informe).

La información contenida en el informe será evaluada de acuerdo a si esta aporta a la comprensión de la solución al problema. La solución debería ser clara y directa, por tanto, los comentarios deberían ser informativos y no extensos.

No puede presentar sólo el informe o sólo el código, se deben entregar ambos.

Compra y copia de tareas

Se ha detectado una red de compra y venta de las tareas semestrales, lo cual constituye una falta de ética grave no aceptada en estudiantes de la Universidad de Concepción.

Es por ello que habrá una exhaustiva verificación de la autenticidad de las soluciones planteadas. Esto significa que los alumnos integrantes de los equipos que no sean capaces de explicar los detalles técnicos de sus soluciones planteadas, tanto a nivel de algoritmos como a nivel de código, serán evaluados con nota mínima obteniendo la reprobación de la asignatura con condición NCR (No Cumple Requisitos).

Recomendación

La tarea puede ser desarrollada con cualquier intérprete usando el sistema operativo que estime conveniente, sin embargo, antes de entregar sus resultados se recomienda asegurar que estos compilen y ejecuten en Jupyter Notebook sobre Windows.

Sobre el Informe

Para confeccionar el informe de la tarea se recomienda seguir las pautas para la confección de un informe de memoria de título para las diferentes carreras en la Facultad de Ingeniería. No obstante lo anterior, a continuación se entregan algunas indicaciones para sus principales secciones:

- **Portada** El informe debe contar con una portada consistente en una página al inicio del informe con la siguiente estructura:
  - en la parte superior derecha (encabezado) debe contener el logo de la Universidad de Concepción con el texto “Facultad de Ingeniería” y en la parte inferior de este el texto “Universidad de Concepción”.
  - en la parte central debe contener el texto “Programación” y en la parte inferior de este el texto “”Tarea Semestral”
  - en la parte inferior derecha debe contener el nombre de todos/as los/as integrantes del equipo, más abajo debe contener el nombre de la carrera de los/as integrantes del equipo.
  - en la parte central inferior (píe de página) debe contener la fecha con el formato “Junio de 2024”

- **Introducción** Una página en que se resuman los aspectos más importantes del trabajo.
- **Organización** Una o dos páginas en las que se identifique las funciones que asumieron como responsables los diferentes integrantes del equipo. Se solicita indicar no sólo el nombre de la función sino una breve descripción de sus principales actividades o responsabilidades. Incluya acá la lista de reuniones sostenidas, medio usado para la reunión, asistentes, objetivo de la reunión, resultado de la reunión. No incluya, ni por compromiso ni por obligación, a integrantes que no hayan trabajado en el proyecto(en caso de interrogación podrían perjudicar su nota).
- **Desarrollo** Esta sección consta de varias subsecciones como por ejemplo:
  - Análisis, cuando se enfrenta al desafío de resolver un problema de mediana o alta complejidad con programación, antes de comenzar con el programa e incluso con la definición de los algoritmos, es necesario analizar cuáles son los datos y procedimientos que se van a implementar. Si los requisitos no están totalmente claros, la solución podría no ser la adecuada y la reingeniería para resolver un problema a este nivel, una vez implementada la solución, tiene alto costo. En definitiva, **el problema a resolver debe estar totalmente claro**.
  - Estrategia, el modelo, el procedimiento, la fórmula, etc. que se utilizarán para pasar a la solución debe especificarse en esta sección. En esta sección pueden aparecer incluso diferentes estrategias de solución las que posteriormente se pueden ir descartando para elegir la mejor, la más factible o la que tendrá menor costo de implementación.
  - Estructuras de datos, permite identificar que variables, arreglos, listas, etc. se requerirán para resolver el problema y qué información se almacenará en cada una de estas estructuras.
  - Algoritmos, permite describir los grandes bloques de funciones que se ejecutarán para resolver el problema. No es necesario describir algoritmos con niveles detallados de los procedimientos ya que esto no resultaría útil para problemas de mediana o alta complejidad.
  - Resultados, En esta sección se puede documentar la ejecución del programa con algunos datos de entrada específicos. La descripción puede venir acompañada de algunos pantallazos correspondientes a la ejecución del programa.
- **Conclusiones** Todo trabajo con las características del exigido en este curso requiere que una vez finalizado los ejecutores reflexionen y describan los aspectos más relevantes de su realización, respuestas preguntas como ¿cumplimos el objetivo del trabajo?, ¿qué aprendimos?, ¿que faltó aprender o resultó más complejo de realizar?, ¿qué decisiones tomamos que facilitaron la realización del trabajo?, etc. puede responderse en esta sección. No se extienda por más de una página en esto.