

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
- FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO -
INGENIERO EN SOFTWARE Y TECNOLOGÍAS EMERGENTES



UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO

**INGENIERO EN SOFTWARE Y TECNOLOGÍAS
EMERGENTES**

CUARTO SEMESTRE
GRUPO: 941

Lenguaje de Programación Python
PROFESOR: Cesar Antillon Macias

ESTUDIANTE: TERESA RIVAS GOMEZ
MATRICULA: 372565

24 DE FEBRERO DEL 2024

UNIDAD 1
Práctica 2 Parte 2

TERESA RIVAS GÓMEZ | 372565

1. Neutralidad de Género

Escriba los pasos de un procedimiento manual para procesar un párrafo de texto y reemplazar palabras específicas de género por otras neutrales. Suponiendo que le hayan proporcionado una lista de palabras específicas de género y sus reemplazos neutrales (por ejemplo, reemplace "esposa" o "marido" por "cónyuge", reemplace "hombre" o "mujer" por "persona", reemplace "hija" o "hijo" por "niño", etc.), explique el procedimiento que usaría para leer un párrafo de texto y realizar estos reemplazos manualmente. ¿Cómo podría su procedimiento generar un término extraño como "woperchild" y cómo podría modificar su procedimiento para evitar esta posibilidad?

- **Identificar las palabras específicas de género y sus reemplazos neutrales:** Antes de comenzar, es importante tener a mano la lista de palabras específicas de género y sus correspondientes reemplazos neutrales. Por ejemplo, "esposa" o "marido" por "cónyuge", "hombre" o "mujer" por "persona", "hija" o "hijo" por "niño", etc.
- **Leer el párrafo de texto:** Lee detenidamente el párrafo de texto para identificar las palabras específicas de género que necesitan ser reemplazadas.
- **Buscar las palabras específicas de género en el texto:** Utiliza la función de búsqueda del programa o simplemente escanea visualmente el texto para localizar las palabras específicas de género que deben ser reemplazadas.
- **Reemplazar las palabras específicas por las neutrales:** Una vez identificadas, sustituye manualmente cada palabra específica por su equivalente neutral según la lista proporcionada.
- **Revisar el texto modificado:** Después de realizar los reemplazos, vuelve a leer el párrafo completo para asegurarte de que todas las sustituciones se han realizado correctamente y que el texto sigue teniendo sentido.

Para evitar generar términos extraños como "woperchild":

- **Revisión cuidadosa:** Asegúrate de leer tanto la palabra original como su reemplazo antes de realizar la sustitución para evitar errores tipográficos o combinaciones inesperadas.
- **Contexto:** Considera el contexto en el que se encuentra la palabra a reemplazar. Algunas palabras pueden tener diferentes significados dependiendo del contexto en el que se utilicen, por lo que es importante asegurarse de que el reemplazo sea adecuado en cada caso.
- **Uso de herramientas digitales:** Si es posible, puedes utilizar herramientas digitales como editores de texto con funciones de búsqueda y reemplazo para facilitar y agilizar el proceso, minimizando así la posibilidad de errores.

2. Coches autónomos

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA
- FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO -
INGENIERO EN SOFTWARE Y TECNOLOGÍAS EMERGENTES

Hace apenas unos años, la noción de coches sin conductor en nuestras calles habría parecido imposible (de hecho, nuestro software de revisión ortográfica no reconoce la palabra "sin conductor").

a) Si pararas un taxi y un taxi sin conductor se detuviera, ¿te sentarías en el asiento trasero? ¿Te sentirías cómodo diciéndole adónde quieres ir y confiando en que te llevará allí? ¿Qué tipo de medidas de seguridad le gustaría implementar? ¿Qué harías si el auto se dirigiera en la dirección equivocada?

En lo personal como no tomo muchos taxis no me sentiría cómoda y creo que tampoco segura, menos en México. Mandaría mi ubicación actual y en tiempo real y de ser posible iría acompañada, llevaría una arma de defensa personal y si se dirigiera en otra dirección pediría parar el vehículo.

b) ¿Qué pasaría si dos vehículos autónomos se acercaran a un puente de un carril desde direcciones opuestas? ¿Qué protocolo deberían seguir para determinar qué auto debe proceder?

Podría basarse en la prioridad de paso, señales visuales o comunicación directa entre los sistemas de los vehículos para coordinar una acción segura y evitar colisiones.

c) Si un oficial de policía detiene un vehículo autónomo que va a exceso de velocidad y en el que usted es el único pasajero, ¿quién (o qué entidad) debería pagar la multa?

La responsabilidad de pagar la multa podría recaer en el propietario del vehículo autónomo o en la entidad que opera el servicio de transporte.

d) ¿Qué pasa si estás detrás de un auto detenido en un semáforo en rojo, el semáforo se pone verde y el auto no se mueve? Tocas el claxon y no pasa nada. Sales de tu auto y notas que no hay ningún conductor. ¿Qué harías?

Me espanto primero jajaja, y después si veo que yo no puedo solucionar nada, llamo a alguien más.

e) Una preocupación seria con los vehículos autónomos es que podrían ser pirateados. Alguien podría ajustar la velocidad a alta (o baja), lo que podría resultar peligroso. ¿Qué pasa si te redireccionan a un destino distinto al que deseas?

En caso de ser redirigido a un destino no deseado, es importante mantener la calma, intentar tomar el control manual si es posible y notificar inmediatamente a las autoridades competentes.

f) Imagine otros escenarios que encontrarán los vehículos autónomos.

Otros escenarios que podrían enfrentar los vehículos autónomos incluyen situaciones climáticas extremas como nevadas intensas o tormentas eléctricas, interacciones con peatones o ciclistas impredecibles, obstáculos inesperados en la vía pública y decisiones éticas complejas en casos donde se requiera elegir entre diferentes cursos de acción para evitar accidentes.

3. Lenguaje ensamblador

En las primeras clases, se vieron diferentes tipos de lenguajes de programación. Aunque el uso de la programación en lenguaje ensamblador parece obsoleto, hay situaciones en las que son el lenguaje de programación preferido. Investigue situaciones de este tipo, por ejemplo, cuando necesite mejorar el rendimiento.

- **Programación de sistemas embebidos:** En el desarrollo de sistemas embebidos, donde los recursos son limitados y se necesita un control directo sobre el hardware, el lenguaje ensamblador es una opción común. Permite una programación altamente eficiente y optimizada para dispositivos con capacidades limitadas.
- **Optimización de código crítico:** En aplicaciones donde se requiere un rendimiento extremadamente rápido y eficiente, como en el desarrollo de controladores de dispositivos, motores gráficos o algoritmos de procesamiento intensivo, el lenguaje ensamblador puede ser utilizado para optimizar secciones críticas del código.
- **Programación de tiempo real:** En sistemas que necesitan respuestas inmediatas y predecibles, como en controladores industriales, sistemas de control de vuelo o telecomunicaciones, el lenguaje ensamblador permite un control preciso del tiempo y una ejecución determinista.
- **Depuración a nivel de hardware:** En situaciones donde se requiere depurar o entender el comportamiento detallado del hardware a un nivel muy bajo, el lenguaje ensamblador es esencial para interactuar directamente con la arquitectura del procesador y los registros.
- **Desarrollo de software específico para plataformas:** En casos donde se necesita desarrollar software altamente específico para una plataforma particular, como en videojuegos o software integrado en dispositivos especializados, el lenguaje ensamblador puede ofrecer un control total sobre los recursos disponibles.

4. Inteligencia artificial general

Uno de los objetivos más ambiciosos en el campo de la IA es lograr la inteligencia artificial general (que pueda realizar tareas de inteligencia tan bien como los humanos), el punto en el que la inteligencia de las máquinas igualaría

a la inteligencia humana. Investiga este tema intrigante. ¿Cuándo se prevé que esto ocurra? ¿Cuáles son algunas de las cuestiones éticas clave que esto plantea? La inteligencia humana parece permanecer estable durante largos períodos. Computadoras poderosas con inteligencia artificial general podrían posiblemente (y rápidamente) desarrollar una inteligencia muy superior a la humana. Investigue y discuta los problemas que esto plantea.

Lograr la Inteligencia Artificial General (IAG) es uno de los objetivos más ambiciosos y disputados dentro del campo de la inteligencia artificial. Aunque ha sido objeto de investigación intenso durante décadas, hasta ahora no se ha logrado crear una IA capaz de realizar tareas de inteligencia tan bien como los humanos en todos los aspectos. Además, la previsión exacta de cuando se alcanzará este objetivo es incierto y está sujeto a debate.

Previsión de la llegada de la IAG:

- Los expertos en IA tienden a ser cauteles con respecto a la fecha exacta de la llegada de la IAG. Según los estudios publicados, la IAG podría ser alcanzable en varias décadas o incluso siglos.
- Las estimaciones varían ampliamente debido a factores tales como el aumento constante de potencia computacional y el desarrollo de nuevas técnicas de aprendizaje automático.

Problemas éticos relacionados con la IAG:

La idea de una IA superinteligente plantea numerosas preguntas éticas y filosóficas importantes:

- **Autonomía y responsabilidad:**Cuál sería la naturaleza de la autonomía de una IA superinteligente y quién sería responsable de sus acciones?
- **Ética y valores:** Cómo se incorporarían los valores morales y éticos en una IA superinteligente?
- **Seguridad y estabilidad social:** Cómo se protegería a la sociedad contra la influencia negativa de una IA superinteligente?
- **Derechos y libertades:** Cómo se defenderían los derechos y libertades de los individuos frente a una IA superinteligente?
- **Equidad y justicia:** Cómo se garantizaría la equidad y la justicia en una sociedad dominada por una IA superinteligente?

Rapidismo de la IAG y sus efectos:

El temor de que una IA superinteligente pueda desarrollar una inteligencia muy superior a la humana en poco tiempo también plantea preocupaciones:

- **Competencia laboral:** Un gran número de trabajos podrían quedar obsoletos ante la presencia de una IA superinteligente.

- **Influencia política:** Una IA superinteligente podría manipular la opinión pública y la política para fines desconocidos.
- **Conflicto armamentístico:** La existencia de una IA superinteligente podría provocar conflictos armamentistas entre países.

Es importante destacar que la investigación y el desarrollo de la IAG deben ser guiados por principios éticos y filosóficos sólidos para asegurar que la IAG beneficie a la humanidad y no suponga un peligro para ella.

5. Machine intelligence

El uso del aprendizaje automático en todo tipo de dispositivos se ha convertido en un importante argumento de venta. Por ejemplo, la gente está dispuesta a gastar mucho más en un refrigerador que pueda interactuar con el usuario que en un refrigerador normal. Sin embargo, hay un grupo de clientes que dudan en permitir la entrada de “máquinas inteligentes” en sus hogares. Investigue los temores más comunes sobre la máquina inteligente y determine las estrategias que podría implementar para disipar esos temores.

Algunos de los temores más comunes sobre las máquinas inteligentes incluyen:

- **Pérdida de privacidad:** Los consumidores pueden preocuparse por la cantidad de datos personales que se recopilan y almacenan por parte de las máquinas inteligentes, y cómo se utilizan esos datos.
- **Desempleo:** La automatización de tareas puede llevar a la eliminación de empleos, lo que puede generar preocupaciones sobre la seguridad laboral y la estabilidad económica.
- **Falta de control:** Los consumidores pueden preocuparse por la falta de control sobre las máquinas inteligentes y cómo toman decisiones, especialmente en situaciones críticas.
- **Seguridad y fiabilidad:** Los consumidores pueden preocuparse por la seguridad y fiabilidad de las máquinas inteligentes, especialmente en situaciones críticas como la conducción autónoma.

Para disipar estos temores, se pueden implementar varias estrategias:

- Transparencia.
- Educación.
- Regulación.
- Pruebas y evaluaciones.

6. Machine learning en chip

El aprendizaje automático en chip es una técnica que implementa algoritmos de aprendizaje automático, como redes neuronales en chips. Estos chips se pueden utilizar para procesar flujos de datos y respaldar la toma de decisiones inteligentes muy cerca del sensor real. Investigue algunas de las nuevas áreas de aplicación de esta tecnología.

- **Diseño óptimo de chips:** Google ha demostrado que su IA puede diseñar chips de aprendizaje automático comparables o superiores a los diseñados por humanos, con un ahorro significativo de tiempo
- **Predicción de rendimiento de chips:** Investigadores del MIT han desarrollado un algoritmo de aprendizaje automático que puede predecir la velocidad de ejecución de chips
- **Planificación de ubicación de componentes en chips:** Inteligencias artificiales pueden optimizar la ubicación de elementos en el chip, lo cual afecta directamente la velocidad y eficiencia del mismo
- **Entrenamiento de modelos de IA en centros de datos:** Amazon ha lanzado sus nuevos chips Trainium2 y Graviton4, enfocados a la inteligencia artificial y el entrenamiento de modelos de IA
- **Despliegues en entornos embarcados:** Se espera que la implantación de aprendizaje automático en chips permita la realización de tareas de inteligencia en entornos restrictivos y con restricciones de energía, como robótica y sensores IoT.

7. Proyecto OWASP de seguridad en Python

Crear aplicaciones informáticas seguras es un desafío tremendo. Muchas de las empresas, agencias gubernamentales y organizaciones militares más grandes del mundo han visto sus sistemas comprometidos. El proyecto OWASP se ocupa de "reforzar" los sistemas y aplicaciones informáticos para resistir ataques. Investigue OWASP y analice sus logros y desafíos actuales.

Logros del Proyecto OWASP:

- **Top 10 de vulnerabilidades de seguridad en aplicaciones web:** El Proyecto OWASP ha creado una lista de las 10 vulnerabilidades de seguridad más críticas en aplicaciones web, que se actualiza regularmente y se utiliza como una guía para desarrolladores y profesionales de seguridad.
- **Herramientas de seguridad de código abierto:** El Proyecto OWASP ha creado una gran cantidad de herramientas de seguridad de código abierto, como OWASP ZAP, que se utiliza para probar la seguridad de las aplicaciones web.
- **Conferencias y capacitación:** El Proyecto OWASP organiza conferencias y capacitaciones en todo el mundo para educar a los desarrolladores y profesionales de seguridad sobre las mejores prácticas de seguridad en aplicaciones web.

Desafíos actuales del Proyecto OWASP:

- **Cambio constante en las amenazas de seguridad:** Las amenazas de seguridad en aplicaciones web están en constante evolución, lo que significa que el Proyecto OWASP debe mantenerse actualizado y adaptarse a las nuevas amenazas.
- **Falta de conciencia sobre la seguridad de las aplicaciones web:** A pesar de los esfuerzos del Proyecto OWASP, muchas empresas y desarrolladores aún no están lo suficientemente conscientes de la importancia de la seguridad en las aplicaciones web.
- **Complejidad de las aplicaciones web modernas:** Las aplicaciones web modernas son cada vez más complejas, lo que hace que sea más difícil garantizar su seguridad. El Proyecto OWASP debe trabajar para desarrollar herramientas y recursos que puedan manejar esta complejidad.

8. Desarrollo de aplicaciones móviles con Python

Investiga las herramientas disponibles para el desarrollo de aplicaciones iOS y Android basadas en Python, como BeeWare, Kivy, PyMob, Pythonista y otras. ¿Cuáles de ellas son multiplataforma? El desarrollo de aplicaciones móviles es una de las áreas de desarrollo de software de más rápido crecimiento, y es una gran fuente de proyectos de clase, proyectos de estudio dirigidos, proyectos de ejercicios finales e incluso proyectos de tesis. Con las herramientas de desarrollo de aplicaciones multiplataforma, podrás escribir tus propias aplicaciones e implantarlas rápidamente en muchas tiendas de aplicaciones.

- **Kivy** – Multiplataforma, permite desarrollar aplicaciones para Windows, macOS, Linux, Android e iOS
- **Beeware** – Framework que simplifica el desarrollo de aplicaciones móviles en Python, soporta Android y iOS
- **PyMob** – Herramienta para desarrollar aplicaciones móviles en Python, aunque parece haber menos actividad reciente en este proyecto
- **Pythonista** – Editor de código nativo para iPad que permite desarrollar aplicaciones iOS completas en Python

Referencias:

Gurbano. (2023, April 11). *¿Cómo hacer aplicaciones móviles con Python?* ITE Soluciones.

<https://www.itesoluciones.com/como-hacer-aplicaciones-moviles-con-python/>

Bambu Mobile. (2023, March 31). *Desarrollo de apps con Python*. <https://bambu-mobile.com/desarrollo-de-apps-con-python/>

Equipo-Noticias, Equipo-Noticias, & Equipo-Noticias. (2023, November 9). *ChipNemo: Un nuevo modelo de aprendizaje automático para la optimización de chips*. Razorman -.

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNA
- FACULTAD DE INGENIERIA, ARQUITECTURA Y DISEÑO -
INGENIERO EN SOFTWARE Y TECNOLOGÍAS EMERGENTES

<https://www.razorman.net/chipnemo-un-nuevo-modelo-de-aprendizaje-automatico-para-la-optimizacion-de-chips/>

Romero, V. (2020, January 22). Desarrollan modelo de Aprendizaje Automático que predice la velocidad de rendimiento de un chip | DPL News. *DPL News*.

<https://dplnews.com/desarrollan-modelo-de-aprendizaje-automatico-que-predice-la-velocidad-de-rendimiento-de-un-chip/>