Sistema Experto: Bored Assistant (BOA)

Expert System: Bored Assistant (BOA)

**Autores: Julian Esteban Giraldo Moncada, Leandro Hurtado Salazar, Luis Miguel Marulanda, Valeria Morales Vanegas**

*Computación Blanda, Universidad Tecnológica de Pereira, Pereira, Colombia*

***Abstract-*** *Our society is obsessed with productivity and time management, BOA is presented as a solution to the ‘not knowing what to do’ that comes up during free time, given the amount of possibilities to choose from to spend our time of leisure. Created with the help of different libraries, BOA recognizes human voice (with speech recognition) and can make searches in differents web pages (with selenium)*

***Resumen-*** *Nuestra sociedad está obsesionada con la productividad y el manejo del tiempo, BOA se introduce como una solución a la indecisión generada en los tiempos libres, dada la cantidad de posibilidades existentes para disfrutar del ocio. Creado con la ayuda de diferentes librerías de python, BOA reconoce la voz humana (con speech recognition) y puede realizar búsquedas en diferentes páginas de internet (con selenium).*

**I. Introducción**

La sociedad en la que vivimos tiene una

obsesión con el tiempo —y su manejo—, hasta el punto en que todo lo medimos con este. Es por esto que aprovechar las veinticuatro horas del día de la mejor manera posible es una preocupación latente en las personas, puesto que la productividad ha pasado a ser la forma en que se mide nuestra contribución a la sociedad.

Esta idea de productividad ha generado un gran

movimiento en redes sociales —y fuera de estas—, que se enfoca en el manejo del tiempo, cómo ser más productivo, entre otros; siempre enfocados en el mayor enemigo: la procrastinación.

Procrastinar es básicamente disfrutar del ocio, hacer cualquier otra cosa con tal de no enfrentarnos a la tarea que nos corresponde. Y es normal, tras largas horas de trabajo o estudio, el cerebro pierde concentración y necesita hacer otras cosas, más aún si la tarea a hacer es considerada aburrida.

El ser humano ha amado el tiempo de ocio

durante toda su historia, fuese usado para hacer más fácil manejar sociedades o como una manera de crear arte y conocimiento —es bien sabido que los grandes pensadores son quienes más tiempo libre suelen tener—; así que enfoquemonos en este tiempo de no productividad, ¿cómo nos gusta pasar ese ‘no hacer nada’? El mundo actual ofrece mil posibilidades: podemos ver vídeos, jugar videojuegos, dedicarnos al arte o al ejercicio, porque el tiempo de ocio no

implica siempre pereza/tirarse en cama. Pero con la obsesión existente de estar siempre haciendo algo productivo llega el dilema —que puede hacerse eterno—, sobre cómo usar el tiempo de descanso, el no hacer nada puede parecer como un vacío imposible de evitar. Así que optamos por abrazarlo y pasar el rato de esa manera, aterrados de nuestra procrastinación pero incapaces de hacer algo para cambiarlo.

Ahora, elegir es algo que a los humanos no se

nos da precisamente bien, y ante las mil posibilidades ofrecidas la impotencia y desesperación pueden hacer bastante daño a nuestro estado mental. BOA es la solución a ese problema.

Asistente Aburrido —Bored Assistant— es una

aplicación que permite al usuario dejar que el

asistente elija por él entre diferentes actividades de ocio, al tiempo que funciona como cualquier otro asistente de búsqueda.

**II. Desarrollo**

Muchos adultos podrían decir que el problema

de la juventud es la constante ‘vagancia’, pero la realidad es la gran cantidad de cosas para hacer cuándo las responsabilidad son mínimas o ya han sido terminadas y la imposibilidad de encontrar qué hacer (que no incluya quedarnos en redes sociales).

Ante este ‘problema’ que enfrentamos de manera constante, la idea de dejar a un programa decidir de forma casi aleatoria las actividades del tiempo de ocio empezó a formarse. Es un concepto sencillo, una app a la que le pides algo para hacer y ella te retorna algo. BOA evolucionó de la aleatoriedad a una IA capaz de reconocer la voz humana y al tiempo se convirtió en un asistente de búsquedas efectivo.

Ahora, dada la necesidad de procesar el lenguaje natural y la decisión de utilizar python como la herramienta para crear el aplicativo, se recurre a la librería de speech recognition que trabaja en conjunto con pyaudio. Pero BOA no solo se dedica a procesar palabras, la librería Selenium es usada para las búsquedas que quiera realizar el usuario, permitiendo a la aplicación abrir cualquier página web que contenga lo buscado.

Siendo más específicos sobre las librerías usadas, estas tienen las siguientes funcionalidades:

● Speech Recognition [1]:

La labor de speech recognition es convertir el habla a texto, en este caso independiente del hablante, puesto que no nos interesa saber quién es la persona que le habla a la aplicación, sino lo que dice.

Al usar esta librería debemos tener en cuenta algunas consideraciones, una es el idioma: inglés es el idioma en el que mejor funciona; afortunadamente la librería trabaja muy bien en

idioma español, lo que nos permite usarla para la aplicación. Otra consideración es que el audio que recibe la librería para procesar debe tener cierto nivel de claridad para reconocer las palabras que se dicen en él, para esto puede ayudar un micrófono que pueda calibrarse con filtro de ruido, o grabar la voz en un lugar silencioso. Speech recognition requiere del uso de pyaudio

● Selenium[2]:

Nos permite controlar un navegador web desde python, con acciones como abrir una ventana del navegador con una página web específica, cambiar la configuración de esta ventana, como tamaño, vista, etc. y tener acceso/manejo sobre el código html de la página web que está abierta en la ventana.

● Tkinter [3]:

Hace parte de las librerías básicas de python, con ella podemos crear un entorno gráfico, con

herramientas de texto, imágenes, botones, barras de desplazamiento, formas y demás para hacer una interfaz de usuario (BOA widget). Teniendo ya el procesamiento del audio con el

que obtenemos texto, a este hay que hacerle otro procesamiento para identificar la acción que quiere realizar el usuario, sea una acción con o sin descripción completa, por esto se refiere a:

● descripción completa: El usuario dice la acción, que puede ser ver, leer, escuchar, hacer, etc. y también el objetivo de la acción, por ejemplo: ver océanos documental. donde ‘ver’ es la acción y ‘océanos documental’ es el objetivo de la acción.

● descripción incompleta: En este puede o no importar la acción, pero el objetivo no importa, la idea del usuario aquí, es que la aplicación escoja el objetivo. Los tipos de descripción se separan en la interfaz con botones distintos para facilitar el filtrado.

De realizarse una descripción completa, el filtro separa la acción del objetivo usando diccionarios de python, identifica estos componentes e identifica también el texto basura, es decir, lo que dice el usuario que no es importante para las búsquedas de la aplicación.

La inteligencia artificial sabrá que se le está hablando cuando en el audio del usuario se detecte la palabra oye. BOA cuenta con una pequeña interfaz similar a un widget, donde encontramos tres botones:

● BOA button (botón superior): Este botón se identifica con la imagen de la aplicación (BOA), que es la silueta de una serpiente. Al presionarlo nos abre una ventana con información sobre los otros botones.

● Botón aburrido (botón del medio): Se utiliza para las búsquedas de descripción incompleta, la aplicación recomienda al usuario qué hacer.

● ¿En qué puedo ayudarte? (Botón superior): Cuando el usuario quiere hacer una búsqueda de descripción completa usa este botón.

Para los dos últimos botones se realiza el

reconocimiento del habla y el procesamiento de

este, una vez identificadas las acciones y/u objetivos BOA se encarga de abrir una ventana de navegador web con la búsqueda de lo que pidió el usuario.

**III. Explicación de código**

La aplicación a nivel de software se divide en

cuatro secciones que son:

● Voice rec: Aquí se hace el reconocimiento de voz.

Primero se graba lo que dice el usuario haciendo uso de speech recognition, con esta misma librería se obtiene el texto del audio grabado, el cuál será la entrada que recibe Voice processing.

● Voice processing:

- Se definen las acciones y objetivos reconocibles por la aplicación.

- Se identifican las palabras no

relevantes(basura) para la búsqueda y se extraen.

- El estado de solicitud se activa si el

usuario le dice 'oye' a BOA.

- Identificamos la solicitud del usuario

extrayendo la acción y dejando el objetivo. Este es el que determina qué tan completa es la descripción del usuario sobre lo que quiere hacer.

● Online:

- La librería de selenium predomina en esta

sección del código, aquí se instancian los

controladores del navegador web, donde

dependiendo de la acción del usuario se ejecutará una función acorde, ejemplo: Si el usuario quiere ver un video de ‘preparación de una torta de zanahoria’, se utiliza la función de videoConcreto(). Lo que hacen esta funciones, es que para cada acción que pueda decir el usuario, se abre una ventana del navegador web, en una página específica acorde a la acción y con el parámetro de búsqueda del objetivo, dependiendo del formato de búsqueda de la página web solicitada. Si la búsqueda no es de carácter concreto, se utilizan las funciones de buscar-acción-(), que se seleccionan de manera aleatoria si el usuario no dice una acción específica,

Ejemplo:

El usuario dice - Oye, quiero hacer algo. O si el usuario dice la acción, pero no el objetivo:

El usuario dice - Oye, quiero leer un artículo. se ejecutaría en este caso buscarArticulo().

- El controlador utilizado para esta versión es chromedriver, que permite utilizar el navegador de google chrome.

● App:

- Todo el proceso de interfaz de usuario y de conección con las secciones ‘Online’, ‘Voice rec’ y ‘Voice processing’ se hace aquí.

- Para la selección aleatoria de las acciones u objetivos a elegir por BOA, se crea una lista de listas, que contienen los valores numéricos asignados a estas características en la sección de ‘Voice processing’.

- Definimos las instrucciones para ‘Botón aburrido’-(botonAburrido) y para ‘¿En qué te puedo ayudar?’-(botonConcreto). En ambos se realiza el proceso de reconocimiento ‘Voice rec’ haciendo uso de la función identificar() para botonAburrido(), y de filtrarsolicitud() para botonConcreto(). Los datos que nos arrojan estas funciones son; acción y/u objetivos, estos serán los parámetros de entrada en ‘Online’.

- Despliega la interfaz de usuario de tres botones.

Durante el desarrollo de la aplicación tuvimos algunos problemas, principalmente con la versión de python que utilizaramos, esto porque speech recognition hace uso de otra librería Pyaudio, que para algunas versiones de python no se encuentra bien empaquetada en los gestores de paquetes. La versión que encontramos con compatibilidad para Pyaudio, sin problemas por dependencia de archivos es python 3.6.3, es con la que se hizo la mayoría del desarrollo, a excepción de partes de la interfaz y del procesamiento del texto, donde no

hubo problema con otras versiones de python. Al hacer uso de Selenium, específicamente del

controlador de chromedriver, la versión de este

controlador debe ser compatible con la versión de chrome que se tenga en la máquina donde se

ejecuta el programa, para este se hicieron pruebas en diferentes máquinas, con versiones de chrome 73, 74 y 75, donde no hubo problemas en la ejecución a excepción de unas máquinas en las que la segmentación por bloques de ejecución de python provocaba que la ventana del navegador se cerrará una vez cargará por completo la información de la página web, la solución de estoes cambiar una parte de la instanciación del controlador web en el segmento de ‘Online’, para que la ventana no se cargará desde dentro de las funciones, sino desde la sección ‘App

**IV. Conclusiones**

El ocio siempre ha sido una de las partes importantes de la actividad humana, puesto que los tiempos libres son los que permiten los mejores descubrimientos, obras de arte y demás (siempre que entendamos tiempos libres como ajenos al trabajo ‘productivo’ que demanda nuestra sociedad), a pesar de que las actividades artísticas se han visto aprovechadas por el capitalismo.

BOA es una herramienta sencilla multipropósito

que permite quitar la culpa generada por el no saber qué hacer y la reemplaza con la certeza de que algo se hará, ayudando también en cualquier búsqueda específica de conocimiento.

**V. Bibliografía**

1. https://pypi.org/project/SpeechRecognition/

2. https://www.seleniumhq.org/docs/

3. https://docs.python.org/3/library/tk.html

4.https://www.google.com/amp/s/blog.laboralkutxa.com/estudio-sobre-el-ocio-a-que-de

dicamos-el-tiempo-libre/amp/